Author:
Payot, Alexandre D. J.

Title:
Shape and Topology Optimisation of External Flows

Shape and Topology Optimisation of External Flows

By

ALEXANDRE DENIS JEAN PAYOT

Department of Aerospace Engineering
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

SCHOOL OF CIVIL, AEROSPACE AND MECHANICAL ENGINEERING

SEPTEMBER 2020

Word count: 68,000

Abstract

The work in this thesis develops methods for aerodynamic topology optimisation (ATO), in particular, and, in general the automatic exploration of physical design spaces. To be suitable for aerospace applications, the framework developed in this work is compatible with boundary fitted meshes; which contrasts with recent methods for ATO which do not maintain an explicit boundary. This makes this work compatible with existing aerodynamic shape optimisation frameworks, and gives those methods the potential to explore radically new designs though topology optimisation.

Development of a parameterisation with a smooth and intuitive geometric response in two and three dimensions is one of the key requirements for TO and the primary focus of this thesis. The new parameterisation, the restricted snakes volume of solid (RSVS), is defined as the profile minimising the length and containing the area specified by a set of volume fractions defined on a grid. This formulation is shown to extend to any layout of volume fractions, and, an analytical equivalence to NURBS is analytically derived. The parameterisation is extended to three dimensions following the natural reformulation of its definition as the geometry minimising area under volume constraints. Like the two dimensional parameterisation, the 3D-RSVS is shown to have intrinsic links to methods in discrete differential geometry.

The new parameterisation is integrated into a modular optimisation framework which supports gradient and agent based optimisation; with geometric, structural and aerodynamic objective functions. The framework is shown to match expected results on the geometric matching of traditional and multi-body aerofoils, the topology optimisation of structural geometries, and, the minimisation of drag on a standard aerodynamic shape optimisation (ASO) benchmark case. This framework outperformed known analytical optima and shape optimisation results on the minimisation of drag at supersonic speeds under area constraints; which have analytical optima, where it successfully explored topology.

Meanwhile flexibility is achieved by using hierarchical design variables both inside the RSVS and through integration with multi-level subdivision optimisation (MLSO). The combination of MLSO and RSVS enables very efficient exploration of complex design spaces, revealing the intricate interactions between geometric and flow topology in supersonic benchmarks. Beyond strict minimisation, this method revealed discontinuous flow behaviours and degenerate multi-modality in the design space. By discovering the properties of the search region, the integrated RSVS-MLSO method shows the potential of flexible automatic design methods to enhance a designers understanding of the physical design space.

Dedication and Acknowledgements

I would like to thank first my advisor Dr. Tom Rendall for his guidance, constant support, and, technical insight throughout the research and writing of this thesis. I am grateful for Prof. Chris Allen's fostering of my research interest during my master research project and his experience and recommendations proved invaluable when time came to present and publish the research in this thesis.

I gratefully acknowledge the funding of the EPSRC for this project and the technical support of the ACRC at Bristol University. Along with my supervisors, I would like to thank Dr Dorian Jones and Nigel Taylor for there continued interest and the stimulating discussions about the relevance of the work and possible developments.

I was lucky to have an office where lively discussion was encouraged, around the twice daily coffee, a pint, or at the cheesecake factory. In particular, Dan and Dom's mentoring at the start of these 4 years helped me effectively navigate our research fields, as well as the conference hotel. Special thanks must go to my colleague Laurence Kedward, with whom the exchange of ideas and our fruitful collaboration kept my interest in our research high until the end.

Completing this thesis took much more than time in the office and would not have happened without the company of my peers Will, Matt, Imanol and Ale, who knew exactly when and how to avoid work discussions during our daily lunches; and my house-mates Will, Helena and Chris for their commiserating during long winter evenings. Along the way I was lucky to meet Teresa, who brings colour to rainy days and who has been a crucial, unwavering support on the journey to complete this thesis. I thank my friends, Amanda, Chris, Hugo, Jessie, Jonny, Robbie, Pippa and Sam, who had the motivation and dedication to show me the invigorating properties of wintry Welsh breezes. I am grateful for Jack and Will who braved the streets of Bristol, exploring, with me, the crafts the city has to offer. Not forgetting those friends further afield who were always there to offer the excuse for holidays away, be it for NYE in Paris, a race down the slopes, or, cultural visits in European cities.

I want to thank my parents, brother, and family; for their support throughout my time studying at the university of Bristol, their belief in me long before that, and their encouragement along whatever paths I have chosen. Finally, I would like to dedicate my thesis to my grandfather, prof. Jean Vincens. From my earliest memory, through his reflections and our conversations on all things cultural, academic and practical; he fostered my budding curiosity and critical thinking. To this day, amongst all the extraordinary people who surround me, he continues to be an example and an inspiration.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

Table of Contents

List of Tables

List of Figures

List of Acronyms

Common acronyms

AI artificial intelligence

ANN artificial neural network

CAD computational aided design

CFD computational fluid dynamics

FEM finite element method

FFD free-form deformation

GPL General Public License

ML Machine Learning

NACA National Advisory Committee for Aeronautics

NASA National Aeronautics and Space Administration

NURBS non-uniform rational B-Splines

POD proper orthogonal decomposition

RMS root mean squared

RMSE root mean squared error

RBF radial basis function

SVD singular value decomposition

UIUC University of Illinois in Urbana Champaign

Acronyms for optimisers and optimisation methods

ACO ant colony optimisation

BFGS Broyden-Fletcher-Goldfarb-Shanno

CG conjugate gradient

DE differential evolution

EA evolutionary algorithm

GSA gravitational search algorithm

GA genetic algorithm

IPM interior point method

KKT Karush-Kuhn-Tucker condition

L-BFGS limited memory Broyden-Fletcher-Goldfarb-Shanno (BFGS)

MDO multi-disciplinary optimisation

MMA Method of Moving Asymptotes

MMO multi-modal optimisation

MOOP multi-objective optimisation problem

NLP non-linear program

PSO particle swarm optimiser

QD quality diversity

QP quadratic program

SALP simplex algorithm for linear programming

SNOPT Sparse Non-linear OPTimizer

SQP sequential quadratic programming

Structural topology optimisation

CSD computational structural dynamics

STO structural topology optimisation

BESO bi-directional ESO

ESO evolutionary structural optimisation

FE finite element

HJ Hamilton-Jacobi

LSF level-set function

LSM level-set method

SIMP solid isotropic material with penalisation

Specialist acronyms

ADODG aerodynamics design optimisation discussion group

ASO aerodynamic shape optimisation

ATO aerodynamic topology optimisation

CMC constant mean curvature

CRM NASA common research model

DNS direct numerical simulation

LES large eddy simulation

MLSO multi-level subdivision optimisation

NS Navier-Stokes

SLIC simple line interface contour

PLIC piecewise linear interface contour

RANS Reynolds-Averaged Navier-Stokes

RSVS restricted snakes volume of solid

3D-RSVS restricted surface volume of solid

r-snake restricted snake

r-surface restricted surface

SPH smoothed particle hydrodynamics

VLM vortex lattice method

VOF volume of fluid

VOS volume of solid

Kulfan's WTT Kulfan's Wind Tunnel Tolerance

NOMENCLATURE

Aerodynamic and Subdivision Symbols

C_A Aerodynamic area constraint.

C_D Drag coefficient.

C_L Lift coefficient.

C_M Moment coefficient.

C_P Pressure coefficient.

$\boldsymbol{\delta}_k$ Subdivision detail vector.

$\Delta y_{BUSEMANN}$ Vertical dimension of a Busemann biplane.

$\mathscr{F}$ Flow solver solution.

M Mach number.

$\mathbf{p}_f$ Re-sampled points projected to the subdivision limit surface.

ϕ_k MLSO basis function.

$\Phi_{\infty,N}$ Matrix projecting points $\mathbf{p}_N$ onto the subdivision limit curve.

$\mathbf{p}_k$ Points at subdivision level k.

$\mathbf{R}_k$ Coarsening operator for a set of points.

$\mathbf{R}_s$ Matrix describing a re-sampling scheme for a set of points.

$\mathbf{S}_k$ Subdivision matrix for refinement.

Optimisation Symbols

f Objective function.

$\mathbf{g}$ Vector function of inequality constraints.

$\mathbf{h}$ Vector function of equality constraints.

$\mathbf{H}_{\mathbf{x}}f$ Hessian matrix of function f with regard to variable vector $\mathbf{x}$.

$\mathscr{L}$ Lagrangian of an optimisation problem.

$\boldsymbol{\lambda}$ Vector of Lagrange multipliers of equality constraints.

$\boldsymbol{\mu}$ Vector of Lagrange multipliers of inequality constraints.

μ Barrier parameter for inequality constrained optimisation.

$\mathbf{x}$ Vector of design variables

$\mathbf{x_0}$ Initial design variables vector.

x^* The star indicates quantity x is taken at an optimum.

$\mathbf{z}_i$ Crossover vector in the DE process.

RSVS Symbols

$\rho_{k,h}$ Height scaling for mode of cell k.

$\rho_{k,j}$ Scaled and adapted response coefficients applied to cell j to generate a smooth RSVS mode from a unit VOS change from cell k.

$\rho_{k,v}$ Volume scaling for mode of cell k.

A Volume of solid required in the restricted snakes volume of solid (RSVS) design cell used in single cell analytical study of the RSVS formulation.

$\mathbf{A(d)}$ Vector of the area inside each design cell and the restricted snake (r-snake).

$\mathbf{a}$ Vector of target areas (a_j) for all design cells.

a_j Volume of solid required in the j$^{\text{th}}$ RSVS design cell.

$A_{S,k}$ Area of triangulated panel k from the r-surface.

c_0 Integration constant for the solution of the RSVS using calculus of variations.

c_1 Integration constant for the solution of the RSVS using calculus of variations.

c_A Area constraint of geometries under aerodynamic optimisation.

C_D Drag coefficient.

C_j Contour of the j$^{\text{th}}$ RSVS design cell.

c_x Horizontal position of the centre of the arc of circle solving the RSVS.

c_y Vertical position of the centre of the arc of circle solving the RSVS.

$\bar{\mathbf{c}}$ Surface centroid based on edge lengths.

δ_a VOS step for gradient calculation of a shape optimisation problem.

$\mathbf{d}$ Vector of the normalised positions d_i of the snaxels.

d_i Normalized position of snaxel along an edge.

e_{feas} Feasibility RMS.

e_{opt} Optimality RMS.

e_{vel} Velocity RMS error.

e_{vf} Volume fraction RMS error.

ε Small stabilising value for the derivatives of the length minimisation objective function.

$\mathbf{F}_i$ Edge vector of the r-snake defined from snaxels i and $i-1$.

$\Delta\mathbf{g}_i$ Direction of travel of a snaxel.

$\mathbf{g_{i,1}}$ Coordinate vector of the snaking grid vertex from which a snaxel originates.

γ_j Vector of polynomial coefficients for the analytical response of the RSVS.

$\gamma_{s,j}$ Vector of polynomial coefficients for the smoothed polynomial response of the RSVS.

i Snaxel index and counter.

j VOS/design/RSVS cell index and counter.

k Step indicator of the SQP process.

l_j Length of RSVS profile in cell j.

λ Lagrange multiplier for the solution of the RSVS using calculus of variations.

M Mach number.

m Number of VOS/design/RSVS cells.

n Number of snaxels in a r-snake/r-surface.

$\mathbf{n_i}$ Outward normal vector to the r-snake on the edge just before snaxel i. $\mathbf{n}$ is defined, for discrete profiles in counter-clockwise order, as the normal corresponding to a -90 degree rotation of the edge vector, i.e. an outward normal with the same length as the edge. For an edge vector $(\Delta x, \Delta y)$, the normal is $(\Delta y, -\Delta x)$.

$\mathbf{p}$ Vector containing the positions of all the snaxels of a r-snake.

$\mathbf{p}_i$ Coordinate vector of a snaxel.

$P_j(x)$ Vector of polynomial coefficients for the analytical response of the RSVS.

$p_j(x)$ Vector of polynomial coefficients for the analytical response of the RSVS.

$q_{k,j}$ Grid adaption coefficient applied to cell j to smooth a unit response in cell k.

$\mathbf{q}_j$ Vector of grid adapted VOS response coefficients to generate a smooth RSVS mode from a unit VOS change in cell j.

r_j Radius of the arc of circle defined in cell j as a result of the calculus of variations.

$\mathbf{R}_A$ Matrix for the calculation of the area of a polygon defined by a vector listing connected vertices.

S 3D Surface for RSVS definition.

Δs_i Length of the edge leading up to snaxel i.

s_e Smoothing coefficient for exact smoothing of the polynomial response.

s_j Smoothing coefficient polynomial response for the j^{th} VOS cell along the profile.

t Step indicator of the r-snake process.

τ_j Coefficient triggering the refinement of a cell.

$\mathbf{v}$ Vector of the normalised velocities v_i of the snaxels.

v_i Normalized velocity of snaxel along an edge.

V_j Non normalised area of cell j (2D-RSVS).

V_j Target volume fraction in cell j (3D-RSVS).

$V_{C_j,k}$ Volume contribution of triangulated panel k from the VOS grid cell j.

$V_{S,k}$ Volume contribution of triangulated panel k from the r-surface.

$\mathbf{V}(\mathbf{d})$ Vector of current volumes contained by the intersection of each VOS cell and the r-surface.

$\mathbf{x}(t,u)$ Parametric definition of coordinate vector of S.

y 2 dimensional geometric profile.

List of Publications

The research presented in this thesis led to the publication of one journal paper, 4 first author conference papers and two second author conference papers.

Breakdown of contributions

Items [S.1] and [C.3] where the results of equal contributions of the two first authors: Laurence Kedward and Alexandre Payot. Presentation for these conference papers were performed by the first author in each case.

Item [S.2] is the result of the publication of Alexander Taylor's master thesis performed under the supervision of C.B. Allen and the mentorship of A.D.J. Payot. This work uses the restricted snakes volume of solid (RSVS) optimisation framework developed in this thesis by A.D.J. Payot, to which A.D. Taylor added a structural solver. Most of the structural results and discussion was written by A.D. Taylor and edited by A.D.J. Payot. Finally the conference presentation was done by A.D.J. Payot.

Journal Articles

[J.1] A. Payot, T. Rendall, and C. Allen, "Restricted snakes volume of solid (RSVS): A parameterisation method for topology optimisation of external aerodynamics," *Computers & Fluids*, vol. 182, pp. 60–84, mar 2019.

Conference Papers

[C.1] A. D. Payot, T. Rendall, and C. B. Allen, "Restricted Snakes: a Flexible Topology Parameterisation Method for Aerodynamic Optimisation," in *55th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2017-1410)*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2017.

[C.2] A. D. Payot, T. Rendall, and C. B. Allen, "Mixing and Refinement of Design Variables for Geometry and Topology Optimization in Aerodynamics," in *35th AIAA Applied Aerodynamics Conference, AIAA AVIATION Forum, (AIAA 2017-3577)*, no. June,

(Reston, Virginia), pp. 1–24, American Institute of Aeronautics and Astronautics, jun 2017.

[C.3] A. D. Payot, L. J. Kedward, T. Rendall, and C. B. Allen, "Optimisation of Multi-Modal Aerodynamic Shape and Topology Problems," in *AIAA Scitech 2019 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2019.

[C.4] A. D. Payot, T. Rendall, and C. B. Allen, "Parametric Surfaces with Volume of Solid Control for Optimisation of Three Dimensional Aerodynamic Topologies," in *AIAA Scitech 2019 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2019.

Second author contributions

[S.1] L. Kedward, A. D. Payot, T. Rendall, and C. B. Allen, "Efficient Multi-Resolution Approaches for Exploration of External Aerodynamic Shape and Topology," in *2018 Applied Aerodynamics Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jun 2018.

[S.2] A. D. Taylor, A. D. Payot, C. B. Allen, and T. Rendall, "Structural Topology Optimisation with R-Snakes Volume of Solid," in *AIAA Scitech 2019 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2019.

CHAPTER 1

INTRODUCTION

Since the Renaissance humankind has strived to understand the world that surrounds us; uncovering rules that allow us to accurately predict natural phenomena. The scientific process and the associated discoveries have driven our societies through accelerating technological change. This progress is most visible through the objects, unimaginable even a few years ago, that now populate our daily lives. As technologies mature and our understanding of the rules that govern them grows; ever tougher requirements are placed on newly designed systems. This increase in complexity forces technology to remain at the frontier of our understanding; with the methods used to apply the latest scientific developments to the design process requiring careful consideration. The tools of engineering must be able to provide accurate and precise information about new designs which may be a radical departure from prior art. The role of these design processes is to turn scientific understanding into actionable information for the realisation of precise goals. NASA's "CFD Vision 2030" [1] highlights challenges that the fluid dynamics community must tackle to enable the next stages of simulation driven design for aerospace applications. This thesis aims to extend the way in which tools used in aerodynamic analysis can be used to explore and generate new designs for existing applications, and be flexible enough to tackle new problems as they arise.

Since the formulation of the Navier-Stokes equations by the eponymous researchers in the nineteenth century [2, 3], progress in fluid dynamics has been focused on the study of these equations. The analytical intractability of these equations is well documented and forms the basis of one of the six remaining unsolved "Millennium prize problems" which reward advances on some of the most important mathematical problems of our

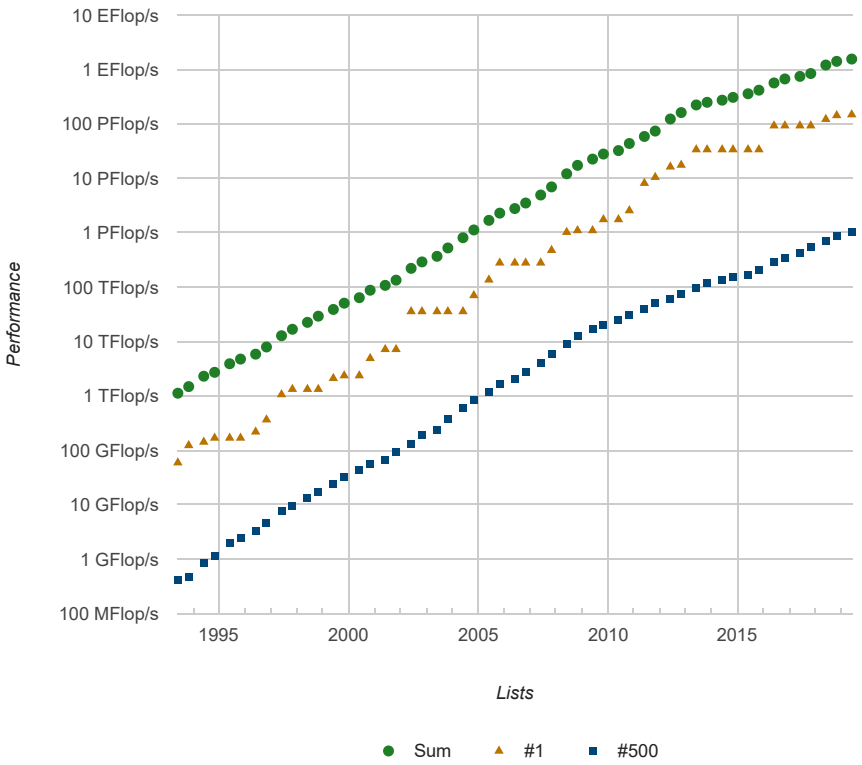


Figure 1.1: Evolution of top 500 HPC systems in the world, showing exponential growth in computing power (Data from TOP500[4]). For comparison most modern smartphones have more than 2GFlop/s.

time. Their complexity has not stopped the extensive use of these equations in science and engineering: by making assumptions and simplifications, analytical solutions and computationally tractable formulations have been derived. These formulations coupled with extensive empirical testing have formed the backbone of aerodynamic design since the 1920s. While successful, these approaches are reaching their limits with new applications and technologies stretching the validity of common underlying assumptions. On the other hand, discrete numerical solutions of the Navier-Stokes equations and high-fidelity simplifications, such as the Euler equations, have shown their capability of accurately predicting complex behaviours but at a large computational expense. This cost has relegated computational fluid dynamics (CFD) approaches to the latter stages of the design process: to validate a final design before prototyping rather than to design its properties. Bringing high resolution tools earlier into the design process has the potential to allow detailed physical behaviours to benefit the performance of new systems. The engineer must possess a high degree of expertise to understand, and successfully exploit, the complexity and non-linearity of physical behaviours to the benefit of the system they

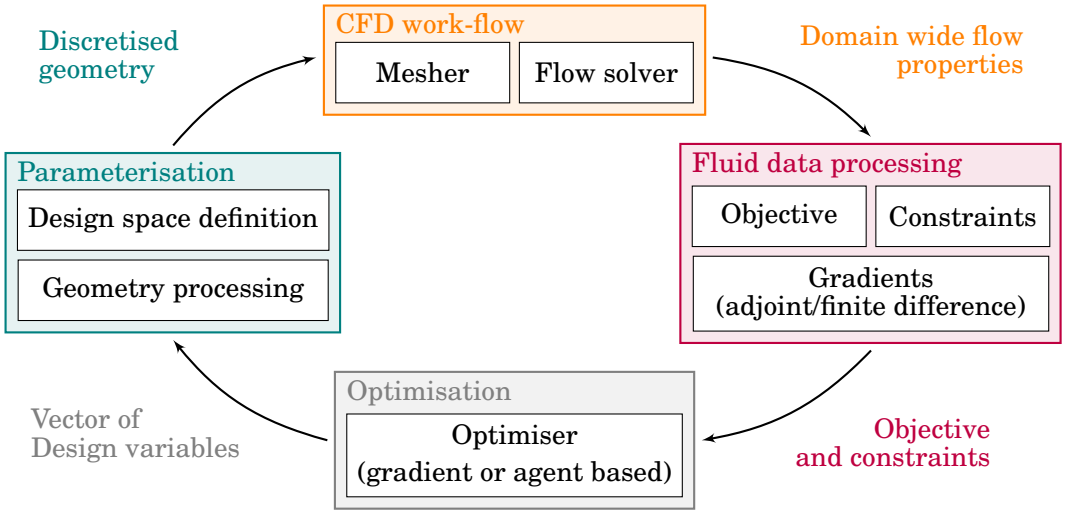


Figure 1.2: Components of a modular aerodynamic shape optimisation framework.

are designing. In this context the development of tools which facilitate the extraction of actionable information from high resolution physical simulations is necessary to allow the next steps in technological advancement.

Increases in computational power (see Figure 1.1) and improvements in CFD tools have created the possibility of using CFD-based optimisation in industrial design. By allowing a systematic and unbiased exploration of a design space, optimisation methods can be used to expand understanding of the problem being tackled, allowing better overall performance. As designers look for further improvements, aircraft manufacturers are turning increasingly to numerical optimisation. Frameworks for aerodynamic optimisation require the integration of mesh generators and flow solvers with optimisation methods through the interface of a parameterisation method. As such the parameterisation method is responsible from translating the numerical values controlled by the optimiser to a geometry which can be meshed, and analysed in a flow solver.

Because of the large effort that goes into the development of efficient and flexible CFD packages, aerodynamic optimisation frameworks tend to follow the modular approach presented in Figure 1.2: integrating established modelling and CFD packages with existing optimisers. The modular approach and the ability of different components of the framework to interface smoothly is critical to ensure compatibility with future methods. Indeed CFD, meshing and optimisation are all fields of research in their own right with new developments permitting improvements in flexibility, accuracy and efficiency. Where a monolithic approach would become rapidly obsolete, the modular approach can be updated with new methods as they are validated without the need to update

(a) A380 Leading edge droop rib, from [6].

(b) APWorks Light Rider motorcycle, from [7].

Figure 1.3: Examples of topology optimised structures (images from ©Altair).

the other components. Current approaches to aerodynamic optimisation have led to the intricate wing profiles seen on the B787 and A330-NEO. Airbus claims fuel savings of 4% for the A330-NEO [5] thanks to a re-engineered wing tip and three dimensional twist optimisation of the wing.

Parameterisation exists to allow an effective interface between optimisers and flow solvers. The complexity of parameterisation arises from the different origins of optimisation methods and CFD processes. Optimisation methods are algorithms devised to find the extrema of functions, and are purely mathematical expressions; while CFD originated from the need to evaluate aerodynamic properties around discretised designs: there is no intrinsic conversion from geometries to the vector forms expected by optimisers. The translation of the mathematical formulations used by optimisers into the geometric designs used by CFD is a complex problem with implications for the efficiency and effectiveness of optimisation frameworks. Parameterisation methods for aerodynamics need to be compact while not artificially limiting the geometric shapes that can be represented [8, 9]. This focus led to aerodynamic optimisation methods capable of efficiently handling small surface changes, using 10s to 100s of design variables in 2 dimensions and 100s to 1000s in 3 dimensions. While the compactness of aerodynamic parameterisations improves the convergence of optimisers, it comes at the cost of handling topological changes.

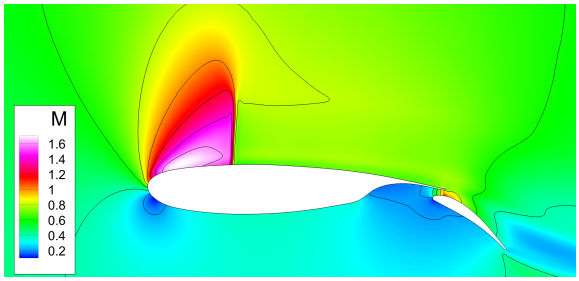
In structural design the benefits of exploring different topologies are key to generating efficient structures. The field of numerical structural topology optimisation (STO) has been an active field of research for the last 30 years and it has recently seen industrial application on the Boeing CH-47 Chinook and the Airbus A380; it allowed a weight

reduction of 17% of underfloor beams compared to a conventional structural optimisation method [10] on the CH-47 and weight reduction of the leading edge droop ribs on the A380 (Figure 1.3). This effort in the finite elements (FE) community has led to parameterisation methods able to represent complex topologies with a homogeneous set of design variables [11–13]. Recent progress in STO has culminated in the numerical optimisation, by Aage et al., of an entire Boeing 777 wing under aerodynamic loads with 1.1 billion degrees of freedom [14]. The algorithm used in that work was capable of building features 40mm long in the 27m half-span, resulting in very detailed internal structures resembling bone and beak structures in nature.

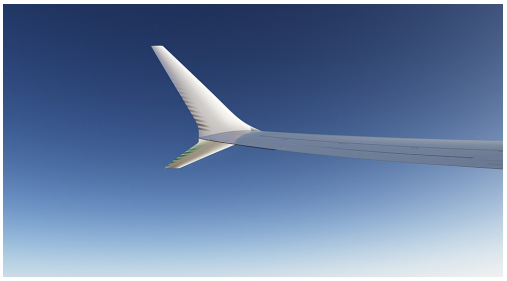
The justification for topological optimisation is straightforward in structural applications, from truss space-frames to honeycomb designs, there are a wide range of possible engineering structures; furthermore the impact of a structural member is readily summarised to boundary interactions. The possibility to reduce designs to a set of external interactions and the Lagrangian formulation of CSD solvers facilitates the implementation of structural topological optimisation within existing designs. The ubiquity of these problems and the flexibility of the existing methods has led to the introduction of STO frameworks inside the widely used SOLIDWORKS [15] and AUTODESK [16] computational aided design (CAD) packages.

There is no such separation in aerodynamics; the aerodynamic shape is intrinsically linked to the rest of the design by its need to be supported by an underlying structure. This means that aerodynamic topological optimisation of an entire aircraft or wing is unlikely to be a reality in the near or medium term. However, there is scope for the aerodynamic topological optimisation of local features; topological optimisation of wing tips would allow feathered or split winglets of the type seen on the MD-11 and Boeing 737-MAX to be explored (Figure 1.4b).

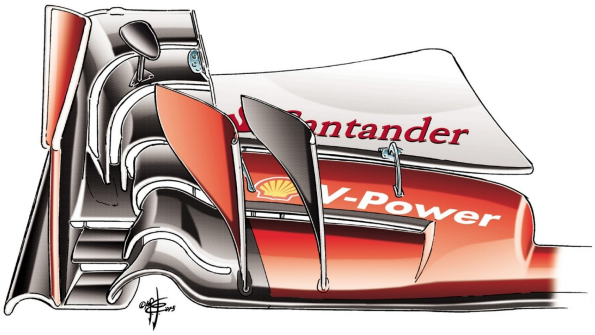
In aerodynamics, topology optimisation is a very new field of investigation. Most progress has been done by adapting the methods of STO to cases where the flow conditions permit the use of Lagrangian solvers. These methods if they use a parameterisation beyond porosity of the fluid domain, use level set definitions with very large number of design variables. While these methods work well for viscous flows with heat exchange, Reynolds numbers are limited in the thousands and compressibility cannot yet be taken into account. These limitations make it impossible to use these solvers and the frameworks that rely on them for the compressible cases common in aerospace vehicle design. Meanwhile none of the current shape optimisation framework for external aerodynamics supports the exploration of topological changes. This is because none of the parameterisa-



(a) C_P plot of a two element aerofoil.



(b) Winglet of the B737-MAX (from Boeing [17]).



(c) Front wing of the SF-15 F1 car (from Giorgio Piola [18]).

(d) Eagle in flight.

Figure 1.4: Examples of complex topology in aerodynamic applications.

tion methods commonly in use in aerodynamic shape optimisation (ASO) can represent different topologies with a homogeneous set of design variables.

An effective topological aerodynamic optimisation framework offers the possibility of radically new designs. This thesis presents the development of a parameterisation method which can handle topology changes while maintaining a compact design space, allowing the exploration of new aerodynamic optimisation problems. Applications to Formula 1, unmanned aerial vehicles, commercial strut-braced wing design and internal engine design could offer significant improvements in the performance of aerospace systems. In an age where climate change is a reality and reduction of carbon emissions a necessity; optimising the efficiency of new aircraft must be a priority.

1.1 Objectives

The aim of this thesis is to develop methods to enable the topology optimisation objects in external, macroscopic flows. There are two main objectives:

1. **Develop an Efficient Topology Parameterisation**

   Parameterisation in two and three dimensions is identified as one of the main bottle necks preventing topological flexibility in optimisation using boundary fitted meshes with CFD. Methods to allow the intuitive parameterisation of topology compatible with traditional modular ASO frameworks are required.

2. **Support Case-agnostic Design Optimisation Processes**

   Automated gathering of information about arbitrary design problems is needed for high fidelity models to support design decisions early during conceptions. Processes and tools should be flexible and generic, not tailored to the specific cases used to validate them. The objective is to demonstrate a capacity to explore and understand a physical design space with as little as possible expert input.

1.2 Outline of Thesis

Chapter 2 presents the background literature to the work in this thesis. In order to build a framework for topology optimisation in aerodynamics a review of optimisation and design space exploration methods is presented . Aerodynamic simulation methods are considered in the context of integration into optimisation frameworks. ASO frameworks are discussed with an emphasis on current aerodynamic parameterisation methods. The methods and successes of STO are discussed as potential routes to follow in the development of aerodynamic topology optimisation frameworks.

A new method for combined shape and topology parameterisation is developed; it is inspired from parametric active contours and volumetric parameterisation methods. The development of this method is presented in Chapter 3 along with some initial parameterisation results highlighting the geometric flexibility of the parameterisation. This method, named restricted snakes volume of solid (RSVS), is analysed and its bases extracted in Chapter 4.

These methods are used within a traditional modular aerodynamic shape optimisation framework in Chapter 5. Gradient based optimisations of transonic and supersonic cases are presented; and results are compared to shape optimisation benchmarks. In Chapter 6 global topological optimisation of supersonic flows is performed, and contrasted with earlier shape optimisation results. An optimisation framework combining the topological flexibility of the RSVS and the very effective multi-level subdivision

optimisation (MLSO) local shape optimisation method is presented and used to explore features of the aerodynamic topological design space.

In Chapter 7, the parameterisation is extended to three dimensions. A robust and efficient, shape and topology marching procedure is developed extending restricted snakes (r-snakes) to be surface objects. The geometric properties and versatility of the parameterisation are shown to carry over in three dimensions. The potential of the restricted surface volume of solid (3D-RSVS), to be an effective and intuitive tool for geometry control for designers and optimisers, is tested.

CHAPTER 2

BACKGROUND

In order to explore the possibility of topological aerodynamic optimisation a review of the following topics is required: Optimisation methods; structural topological optimisation; computational fluid dynamics (CFD) methods in optimisation; and, aerodynamic optimisation frameworks.

2.1 Optimisation Methods

Many methods for optimisation have been developed in the past 60 years and the advent of computers has revolutionised the field, transforming it from a purely mathematical endeavour to an algorithmic one. This section aims to provide an overview of the main mathematical optimisation methods as well as some of the challenges facing the field, especially regarding the practical usage of optimisation for design. The choice of an optimisation method is entirely dependent on the problem that needs to be solved, the properties that drive that choice are explored here.

2.1.1 Terminology and Key Concepts

In general an optimisation problem is formulated as the minimisation (or maximisation) of one or multiple *objective functions* (f) under equality ($\mathbf{h}$) and inequality ($\mathbf{g}$) *constraints*. The objective function and the constraints vary over a *design space* described by *design variables* ($\mathbf{x}$) best represented in vector form. Objective functions can be very easy to compute or require computationally expensive programs as is the case in aerodynamics. In agent based optimisation the objective function and the constraint violation are often

referred to as the *fitness* of a solution. The canonical form of optimisation problems is presented in Equation 2.1; in this formulation no restriction is placed on the smoothness, linearity and continuity of any of the variables and function.

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{subject to}: \quad & \mathbf{g}(\mathbf{x}) \geq \mathbf{0} \\ & \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned} \tag{2.1}$$

Optimal solutions, also known as optimisers or optima, are denoted in the field with a superscripted asterisk: f^* refers to the optimum objective value, while $\mathbf{x}^*$ marks the best design variables. An optimum is defined as a point which solves the *necessary* and *sufficient* conditions of the case; these conditions depend on the properties of the objectives and constraints (smoothness, continuity, differentiability).

The formalism of optimisation is very flexible and adopting its formulation can provide insight into a wide range of practical problems. When applied to observations of nature the formulation of the correct optimisation problem allows the precise modelling of biological behaviours; this is notably the case for trees which grow in shapes of constant surface stress as was investigated by Mattheck and Burkhardt [19]. Optimisation saw one of its first applications in the efficient assignment of resources in the US Army during the second world war [20]; since then it has seen widespread use in economics and engineering.

Optimisation is applicable to a wide range of problems not usually thought of as optimisation; notably parameter fitting and solvers. One example of this applicability across a range of problems is the derivation of the solution to least squares regression problems; this can be solved as a root finding problem [21] or an optimisation problem [22] . Recently, the use of optimisation for model tuning is most visible in the extensive use of back-propagation [23] for the training of artificial neural network (ANN) in the field of artificial intelligence. Physical solvers can also be understood as very specialised optimisers minimising the residual error of the system of equations which they were designed for. This alternate formulation of the physical system of equations can provide the tools required to understand behaviours of the system by describing them using calculus of variations. This wide variety of problems means that many different types of methods classify as optimisers, the choice of which being governed by the structure of the mathematical program in general, and the properties of the objective and constraints in particular.

Optimisers are separated in the two categories of global and local optima. The *global optimum* refers to the overall best solution to an optimisation problem. A *local optimiser* is a solution that is strictly better than solutions immediately next to it, this concept only exists for real valued design variables as it requires continuity of the design space to be evaluated. One of the critical properties of an objective function in the context of optimisation is its *modality*, this concept refers to the number of local optima that a function possesses. A *multi-modal* optimisation problem has multiple local minima which makes it significantly harder to optimise than a *convex* problem. Modality depends mostly on the objective function but the choice of design variables and constraints can also have a significant impact.

Optimisation methods are classified into local and global optimisation methods depending on the type of optimisers they can find. Local methods use a procedure to generate steps that will always improve the objective function value; this leads to an optimiser that will depend on the start location. The algorithm will be unable to move to worse solutions to explore regions further away. A global optimisation method will include techniques to allow it to cross regions of the design space which are worse in order to find the global optimiser. Modality is only one of the properties affecting the choice of an optimisation method; objective function smoothness, constraint linearity and design space continuity all affect the performance of various optimisation methods.

2.1.2 Local Optimisation Methods

This section presents a selection of available local optimisation algorithms. First, gradient-free methods are presented, these were historically the first optimisers to be used, but still see use on specific problems to which they are particularly well suited. Then, first order gradient methods of steepest descent and conjugate gradient are introduced as more efficient methods for smooth objective functions. The use of second derivatives in Newton and Quasi Newton methods is discussed in Section 2.1.2.4. Constraint handling options for local optimisers are presented including the sequential quadratic programming (SQP) approach for non-linear programs. Throughout the section optimisers which can tackle non-linear optimisation problems are tested on the Rosenbrock banana function [24], the formulation of this function is presented in Appendix A.1. This function is a standard test case of an analytical function displaying a difficult to navigate “valley” which illustrates pit falls of some gradient based optimisers. Implementation of the optimisers for this review is that of the python package *scipy*[25], more detail is provided in Appendix A.2. Finally a convergence comparison of all these algorithms is presented

Algorithm 1 Golden section search algorithm.

Let $x_l = a$ and $x_u = b$ be the lower and upper bound of interval $[a,\ b]$ defined in Equation 2.2
Let x_i be a point inside the interval distance l from the lower bound and distance u from the upper bound.
With $u/l = \left(1+\sqrt{5}\right)/2$ convergence on the optimum is most efficient.
while $x_u - x_l > \epsilon$ where ϵ is a user defined convergence tolerance. **do**
  $x_t = x_l + (x_u - x_i)$
  Next *replacing the appropriate bound with the worst performing of* x_i *and* x_t.
  if $x_i < x_t$ **then**
    if $f(x_i) < f(x_t)$ **then** $x_u = x_t$ **else** $x_l = x_i$ and $x_i = x_t$ **end if**
  else
    if $f(x_i) < f(x_t)$ **then** $x_l = x_t$ **else** $x_u = x_i$ and $x_i = x_t$ **end if**
  end if
end while

on the N-dimensional extension of the Rosenbrock function in Section 2.1.2.7.

2.1.2.1 Univariate optimisation

The simplest type of optimisations to tackle are univariate uni-modal unconstrained problems. These cases are usually expressed on a closed interval $[a,\ b]$ of the objective function on which it is expected to be uni-modal. The formulation is as follows:

$$\min_x f(x) \quad \forall x \in [a,\ b] \tag{2.2}$$

In these cases finding minimisers is closely related to the process of finding the roots of a function and methods which were originally developed for root-finding can be adapted to find minimisers. This is the case of bisection search which is used to find roots by bracketing the solution between two progressively closer abscissae. For optimisation this becomes the Golden-section search [26] which maintains a triplet of points progressively bracketing the minimum in the interval $[a,\ b]$. The algorithm for this search method is presented in Algorithm 1.

While simple and limited in scope this method is still extensively used to find optimal parameter settings; because it requires no gradients it is easily applicable to black-box systems. It is also commonly used to perform line-searches in optimisation in multidimensional design spaces; once a search direction has been defined a univariate method searches along that direction for a minimum.

2.1.2.2 Multivariate gradient-less optimisers

Two early multi-variate optimisation methods stand out for their continued use in research and teaching. The simplex algorithm for linear programming (SALP) was developed after the Second World War by Dantzig to resolve the problem of assignment of resources [27]. Linear programs are a special kind of optimisation problem where the objective and constraints are linear combinations of the design variables. The formulation of a linear program is shown in Equation 2.3 in which $\mathbf{A}$ is a matrix of real coefficients, and, $\mathbf{b}$ and $\mathbf{c}$ are real valued vectors.

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to}: \quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} \end{aligned} \tag{2.3}$$

Linear programs refer to the subset of optimisation problems that can be represented in the form above. These include practical problems such as assignment of resources and portfolio optimisation [28]. Linear programs are best understood in geometric terms; each inequality constraint separates the design space in half and a set of coherent constraints will create a convex polytopal feasible region. Since the objective function is linear, the optimum is known to lie at one of the vertices of this polytope. *The Simplex Method for Linear Programming* (SMLP) developed by Dantzig in the 1940s [20] follows the edges of the constraint polytope from vertex to vertex until further rotation cannot yield an improvement in objective function. Later algorithms known as barrier methods and *interior point methods (IPM)* improved upon the simplex algorithm resolving issues of degeneracy and exponential computational efforts [29] by guaranteeing solutions in polynomial time. Interior point method (IPM)s work by crossing the feasible part of the design space to approach the correct vertex or plane of the convex design space. IPMs will be discussed in more detail as they can be adapted to use on non-linear constrained optimisation problems.

The second gradient-less optimisation method considered here is the Nelder-Mead algorithm developed by the authors of the same name in 1965 [30]. This algorithm was designed to solve unconstrained problems on non-smooth functions without the need for gradients. The formulation makes it easy to implement in n-dimensions. It is often used in parameter estimation from noisy information thanks to its gradient less formulation, notably in medicine and chemistry [31]. This algorithm works by marching a convex simplex progressively towards better solutions. A simplex is a convex polytope with $n+1$ vertices in n-dimensions, i.e. the simplest polytope in any dimension (triangle in 2-dimensions, tetrahedron in 3). The iteration is carried out in three stages described

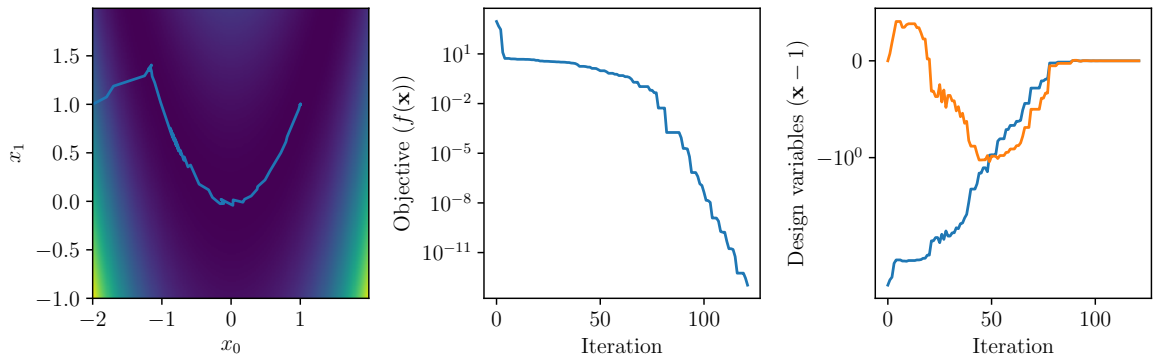


Figure 2.1: Convergence of the Nelder-Mead algorithm on the 2D-Rosenbrock function.

Algorithm 2 Basic step of the Nelder-Mead algorithm for a minimisation problem.

1: Order simplex vertices according to objective function value ($f(\mathbf{x_l}) < f(\mathbf{x_i}) < f(\mathbf{x_u})$)
2: Calculate the centroid of the best side (edge $\mathbf{x_l}$-$\mathbf{x_i}$)
3: Generate a new vertex ($\mathbf{x_r}$) by reflecting the worst vertex through the centroid

in Algorithm 2. Figure 2.1 shows convergence of the algorithm on the 2D Rosenbrock function.

The two algorithms presented in this section show how simple procedures relying on very little information can be used to progress towards an optimum. While these methods are still used in some fields, more complex and efficient approaches are needed for aerodynamic optimisation. Indeed SMLP relies on the formulation of a problem into a linear program which is not possible for non-linear aerodynamics problems. The Nelder-Mead algorithm has however been combined with a Genetic Algorithm (GA) (c.f. Section 2.1.3.2) as part of a hybrid optimisation method [32], further discussion of hybrid methods is presented in Section 2.1.3.3. For local optimisers, the additional information provided by gradients enables much more efficient search for smooth objective functions, especially with many dimensions. The impact of dimensionality on the Nelder-Mead algorithm compared to gradient based optimisers is presented in Section 2.1.2.7.

2.1.2.3 Steepest descent and conjugate gradient

Up to now the algorithms studied have only used the objective function and constraints value to choose the next iterate of the optimisation method. The algorithms in this section represent a step change in approach as they use the first derivative to choose a more direct path to the optimum. For gradient based optimisation to be a good choice the objective function must be smooth and differentiable. The simplest gradient based method

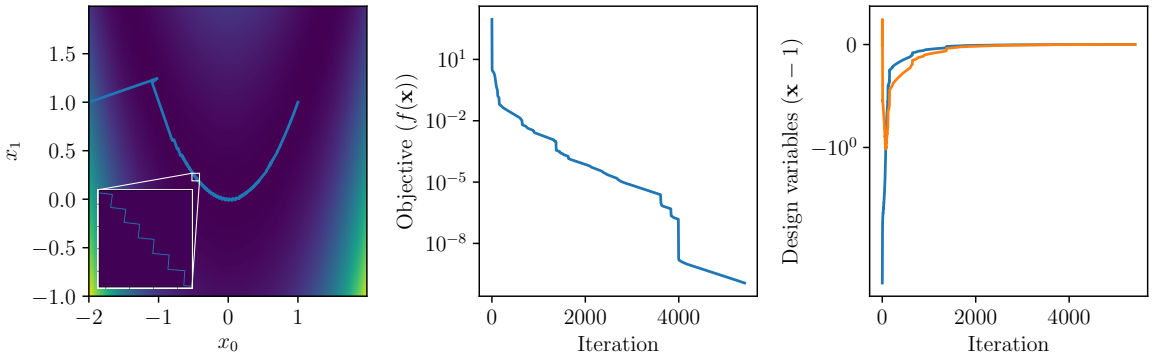


Figure 2.2: Convergence of steepest descent on the 2D-Rosenbrock function.

is the steepest descent algorithm. It was originally proposed by Cauchy to numerically solve simultaneous systems of Equations [33] with an immediate application to fitting equations to astronomical observations. Later adapted for non-linear minimization by Curry [34]; steepest-descent is a two step algorithm: first it uses the gradient of the objective function as the step direction for the next iteration and then performs a line-search along that direction. This two step process forms the basis of all gradient based optimisation: algorithms differentiate themselves by how a search direction is calculated. For an unconstrained problem using the notation of Equation 2.1, the step direction at iteration $t+1$ is calculated using 2.4.

$$\Delta \mathbf{x}_{SD}^{t} = -\nabla_{\mathbf{x}} f\left(\mathbf{x}^{t}\right) \tag{2.4}$$

$$\mathbf{x}^{t+1} = \mathbf{x}^{t} + \alpha^{t}\, \Delta \mathbf{x}_{SD}^{t} \tag{2.5}$$

Once the search direction $\Delta \mathbf{x}^{t+1}$ has been established, the length of the step α^{t+1} must be found. This is done by performing a 1-dimensional optimisation along that search direction. Many methods have been developed to perform line-searches, some are exact (bisection or golden section searches) while others are only designed to find an approximate minimum along that line. The amount of work of the line-search depends on the cost of the objective function and the dimensionality of the design space: if computing the search direction is very expensive the line-search will need to be precise to exploit the maximum of that information. Conversely if the direction is cheap to evaluate an approximate line-search will be sufficient. Wolfe [35, 36] studied the minimal requirements for the value of the step length α^{t+1} to guarantee the convergence of unconstrained gradient based algorithms on a stationary point of the objective function.

To this day the requirements set out in [35, 36], now known as the "Wolfe condition", form the basis of line search algorithms.

Steepest descent is rarely used for optimisation because it has undesirable convergence behaviour. At the end of an exact line-search the gradient along the search direction is 0, this implies that the gradient at the new point will be normal to the previous search direction. This causes a zigzagging search path which leads to slow convergence. This behaviour is very prominent on the Rosenbrock function and can be clearly seen in Figure 2.2. This effect is so pronounced that the convergence is in fact slower than for the Nelder-Mead algorithm (Figure 2.1). This method was used in aerodynamic design optimisation in conjunction with gradient smoothing by Jameson et al. [37] to validate optimisation procedures based on the adjoint method.

The steepest descent algorithm can be improved by using the method of conjugate gradients developed in 1952 by Hestenes and Stiefel [38]. Originally used as a direct method to find solutions to linear systems of n equations in at most n steps; it is also used as an iterative method for many smooth, convex optimisation problems of low dimensionality. The conjugate gradient method uses the gradient information to generate a search direction conjugate to all previous search directions. This is achieved using Equation 2.6 by adding the previous step direction weighted with a parameter β^t. There are a number of formulae for this parameter of which the most well-known are: the Hestenes-Stiefel (HS) [38] (Equation 2.7); Fletcher-Reeves (FR) [39] (Equation 2.8); Polak-Ribière (PR) [40] (Equation 2.9) and Dai-Yuan (DY) [41] (Equation 2.10) formulations.

$$\mathbf{x}^{t+1} = \mathbf{x}^t + \alpha^t \Delta\mathbf{x}^t_{CG} \qquad \text{with}: \quad \Delta\mathbf{x}^t_{CG} = \Delta\mathbf{x}^t_{SD} + \beta^{t-1}\Delta\mathbf{x}^{t-1}_{CG} \tag{2.6}$$

$$\beta^t_{HS} = -\frac{\nabla_{\mathbf{x}} f\left(\mathbf{x}^{t+1}\right) \cdot \left(\nabla_{\mathbf{x}} f\left(\mathbf{x}^{t+1}\right) - \nabla_{\mathbf{x}} f\left(\mathbf{x}^t\right)\right)}{\Delta\mathbf{x}^t_{CG} \cdot \left(\nabla_{\mathbf{x}} f\left(\mathbf{x}^{t+1}\right) - \nabla_{\mathbf{x}} f\left(\mathbf{x}^t\right)\right)} \tag{2.7}$$

$$\beta^t_{FR} = \frac{\left\|\nabla_{\mathbf{x}} f\left(\mathbf{x}^{t+1}\right)\right\|^2}{\left\|\nabla_{\mathbf{x}} f\left(\mathbf{x}^t\right)\right\|^2} \tag{2.8}$$

$$\beta^t_{PR} = \frac{\nabla_{\mathbf{x}} f\left(\mathbf{x}^{t+1}\right) \cdot \left(\nabla_{\mathbf{x}} f\left(\mathbf{x}^{t+1}\right) - \nabla_{\mathbf{x}} f\left(\mathbf{x}^t\right)\right)}{\left\|\nabla_{\mathbf{x}} f\left(\mathbf{x}^t\right)\right\|^2} \tag{2.9}$$

$$\beta^t_{DY} = \frac{\nabla_{\mathbf{x}} f\left(\mathbf{x}^{t+1}\right) \cdot \Delta\mathbf{x}^{t+1}_{CG}}{\nabla_{\mathbf{x}} f\left(\mathbf{x}^t\right) \cdot \Delta\mathbf{x}^t_{CG}} \tag{2.10}$$

These formulations allow convergence on an optima in only 10s of iterations and function calls on the Rosenbrock function (Figure 2.3) compared to 1000s for the steepest descent. The convergence of each of these formulations has been extensively studied [41]. For non-quadratic objective functions each formulation leads to different performance

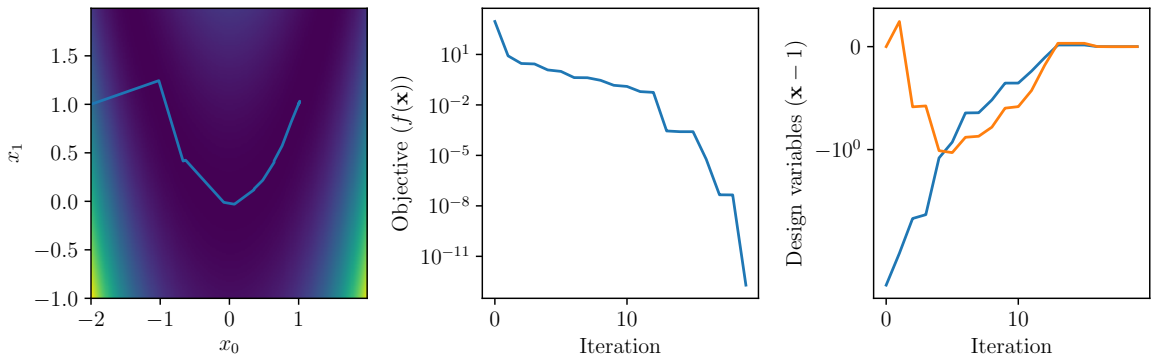


Figure 2.3: Convergence of a conjugate gradient methods with the Polak-Ribière (PR) [40] (Equation 2.9) scaling on the 2D-Rosenbrock function.

and some of the most efficient modern conjugate gradient algorithms dynamically change between formulations [42]. As was the case for the steepest descent method the step length is determined by a line-search along the direction defined by $\Delta\mathbf{x}_{CG}^{t+1}$. While the choice of a line-search is not considered in further detail in this review it is a significant problem, especially in practical cases of optimisation. The computational cost of the line-search can be as large or greater than that of the gradient, especially when fast methods for the computation of derivatives are available.

While the conjugate gradient method does not naturally support the enforcement of constraints, algorithms which use conjugate directions have been developed to do so. Constraint handling methods for non-linear gradient based optimisation are discussed in Section 2.1.2.5.

Steepest descent was used in aerodynamic optimisation by Jameson et al. in a proof of concept of their adjoint gradient calculation [37], however almost all other implementations of first order gradient based schemes have relied on conjugate gradient. These methods have seen significant use in Aerodynamic optimisation due to their simplicity of implementation and their robustness to poor quality gradients [43]. However, in applications where the evaluation of the objective function is expensive, higher order methods are often preferred. Conjugate gradient methods are commonly used in machine learning with back-propagation to optimise the weights of ANNs. Higher order methods would be too expensive in light of the number of design variables (one per neural network weight) which is needed for these applications.

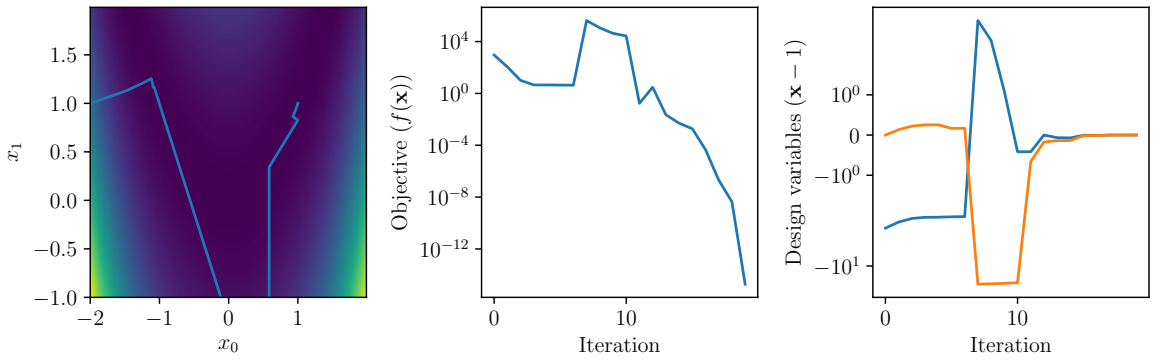
2.1.2.4 Newton and quasi-Newton methods

While first order gradient descent methods are a significant improvement on the Nelder-Mead algorithm for non-linear optimisation of smooth objective functions, second order methods can lead to a much faster convergence. The first method of this type was developed by Newton in 1671 [44]. The Newton method for optimisation (closely related to the Newton-Raphson method used for root finding) uses a formulation which iteratively looks for roots (zeros) of the gradient. The Hessian can be seen as providing optimal scaling of the steepest descent direction [44]. The expression for the step of the Newton method is presented in Equation 2.11.

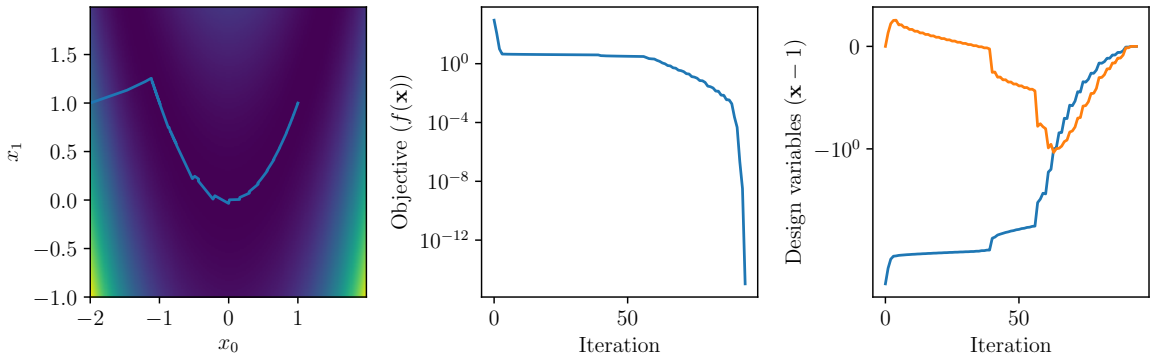
$$\Delta \mathbf{x}_N^{t+1} = -\left(\nabla^2 f\left(\mathbf{x}^t\right)\right)^{-1} \nabla f\left(\mathbf{x}^t\right) \tag{2.11}$$

The convergence performance of the Newton method 'close' to the optimum displays super-linear convergence. However far from a local optimum on highly non-linear functions the convergence is erratic; contrary to previous algorithms the step direction is not guaranteed to be a descent direction. This erratic behaviour is clearly visible in Figure 2.4a with the design variables and objective jumping away from the optimum before converging in very few steps. Unlike previous methods the classical Newton method does not involve a line-search, however other methods (called damped Newton) call for a step scaling $\alpha^{t+1} \in [0,\ 1]$. When using a line search the Newton algorithm converges very slowly through the far-field as can be seen in Figure 2.4b and summarised in section 2.1.2.7. For many practical applications, evaluations of the Hessian matrix and its inverse is impractical: computational cost and ill-conditioning of the Hessian render the use of the Newton method impossible.

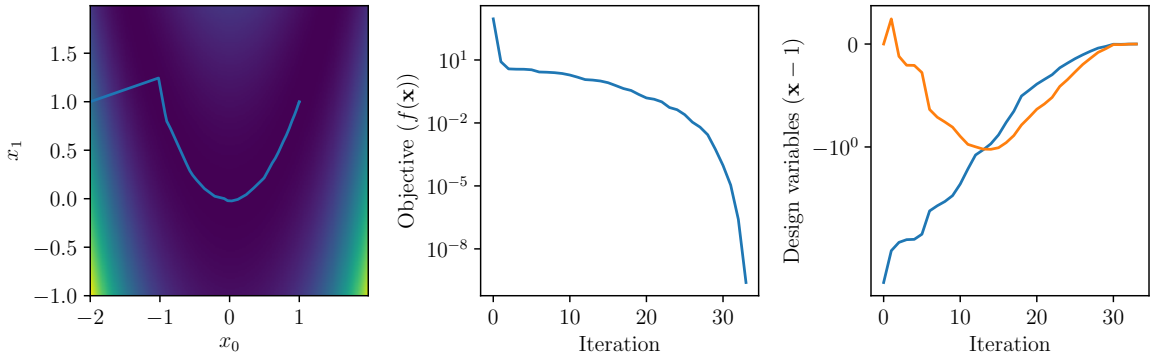
These limitations have led to the development of a number of Quasi-Newton methods which rely on approximations of the Hessian and include safeguards to improve far-field convergence. Approximations of the Hessian leverage the convergence behaviour of second order algorithms without the cost of evaluating the matrix of second derivatives. Some of the most used methods are the Davidon-Fletcher-Powell (DFP) method, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) and the limited memory BFGS (L-BFGS) [45]. As is the case for the Newton method, the performance of these algorithms has been shown to be locally super-linearly convergent [45, 46] which makes them very efficient close to a local optimum. Newton style methods are generally limited in their number of design variables by memory and storage requirements, as an approximation of the inverted Hessian needs to be stored. These methods have seen a lot of



(a) Convergence of the Newton algorithm.



(b) Convergence of the Newton algorithm with line search.



(c) Convergence of the Newton algorithm with a BFGS approximated Hessian.

Figure 2.4: Convergence of exact Hessian Newton algorithms with and without line search, and of a Newton algorithm with a BFGS approximation on the 2D-Rosenbrock function.

use in aerodynamic optimisation as part of SQP implementations discussed in the next paragraphs.

2.1.2.5 Constraint handling for non-linear problems

The Newton and related quasi-Newton methods were designed as unconstrained optimisation algorithms, however most practical problems involve equality and/or inequality constraints. To take an example from aerodynamics, an aerofoil optimisation will always have either a minimum thickness or volume constraint to avoid the reduction to a flat plate. The discussion of gradient based optimisers up to this point has been limited to unconstrained problems. This initial focus mirrors the development of optimisation algorithms and their progressive adaptation from solvers for specific systems of equations to iterative processes for the exploration of arbitrary optimisation problems. As is the case for most aspects of optimisation, constraint handling is its own active field of research with approaches designed for problems with very different problem sizes.

There are two aspects to handling constraints in non-linear programs (NLPs): far-field progression, and, convergence in the neighbourhood of an optimum solution. Both of these aspects require careful handling of the constraint to ensure the final design is optimal and feasible. Significant study has gone into establishing the convergence properties of Newton and conjugate gradient formulations; proving convergence speed under the assumption of a quadratic problem. These concerns have guided the implementation of constraint handling frameworks which would impact convergence properties as little as possible. The main way in which is achieved is by transforming constrained optimisation problems into equivalent unconstrained equivalents. The constraints are included in the objective and the optimisation program is reformulated in terms of the original design variables and additional variables representing how close the current solution is to constraint violations. The focus in this subsection is on the two main approaches which have been used for large scale non-linear programming problem, in industry and academia: interior point and active set methods.

Barrier methods were the precursors to interior point methods (IPMs) and form the basis of their implementation. An in-depth review of these methods is available Forsgren et al. [47]. Barrier methods rely on the transformation of inequality constrained minimisation problems into an unconstrained problem. The inequality constraints are replaced by one-sided diverging functions which remain smooth over the “interior” of the feasible design space. The best performing barrier methods use natural logarithmic barriers [47]; the general non linear program of Equation 2.1 is transformed into its barrier method

formulation below (Equation 2.12). The effect of the barriers are controlled by parameter μ which as it tends to 0 the optimiser of the unconstrained program will tend to an optimiser of the original constrained NLP.

$$\begin{aligned} &\min_{\mathbf{x}} \quad f(\mathbf{x}) \qquad \text{s.t.:} \quad \mathbf{g}(\mathbf{x}) \geq 0 \\ &\min_{\mathbf{x}} \quad f(\mathbf{x}) - \mu \sum_{i=1}^{m} \ln(g_i(\mathbf{x})) \end{aligned} \tag{2.12}$$

These type of barrier methods are not in regular use today as they exhibit ill-conditioning and poor behaviour as the value of μ is reduced [47]; primal-dual interior point methods build on their foundations to permit efficient handling of inequality constraints. The name of the method comes from its simultaneous use of the primal variables (design variables $\mathbf{x}$) and the dual variables (Lagrange multipliers λ). Those constraint application methods alter the objective function and, as a consequence, the position of the optimum in such a way that it can never lie on the constraint. The barriers always have some effect, meaning that the optimum to the original constrained problem can only be found in the limit as μ tends to 0 and the barrier becomes discontinuous.

By contrast active set methods work by separating constraints into *active* and *inactive* and treating them differently. The active constraints are treated as equality constraints, while inactive constraint are removed from the optimisation at the given step. The simplex algorithm for linear programming discussed in Section 2.1.2.6 is an example of an active set method. The most common active set method for non-linear programs is the SQP method first presented by Wilson [48]. Before a new iterate can be defined active set methods need to find the set of constraints that are violated, which can be an expensive combinatorial procedure if there are many non-linear inequality constraints. One of the big benefits of the SQP method is that, unlike barrier methods, no feasible point is required [49]; finding such a point can be extremely expensive in its own right. Due to its ubiquity in aerodynamic optimisation SQP methods are explored in more detail in the following section (Section 2.1.2.6).

It is possible to combine both interior point and SQP methods into a single optimiser letting each handle the constraints they are best suited to [47]. Alternatively equality constraints are applied on interior point methods using a penalty function. The quadratic norm penalty function is a popular choice; it creates a convex descent direction towards feasible values of the equality constraint.

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \qquad \text{s.t.}: \quad \begin{matrix} \mathbf{g}(\mathbf{x}) \geq 0 \\ \mathbf{h}(\mathbf{x}) = 0 \end{matrix} \\ \min_{\mathbf{x}} \quad & f(\mathbf{x}) - \mu \sum_{i=1}^{m} \ln(g_i(\mathbf{x})) + \frac{1}{2\mu} \|\mathbf{h}(\mathbf{x})\|^2 \end{aligned} \tag{2.13}$$

2.1.2.6 Sequential quadratic programming

Aerodynamic optimisation problems present a number of challenges in terms of their non-linearity and the type of constraints that may be applied. The number of non-linear inequality constraints is usually small and finding a feasible point expensive making the SQP method a very effective choice. While a comprehensive explanation of SQP algorithms is beyond this section, the interested reader is referred to the excellent review of Boggs and Tolle [49]. The SQP class of methods approximates the Non-Linear Program (NLP) at the current point using a Quadratic Program (QP); the optimiser of this quadratic approximation provides the next iterate of the SQP algorithm. The QP approximating a general NLP is presented in Equation 2.14.

$$\begin{aligned} \min_{\mathbf{d_x}} \quad & \mathbf{d_x}^T B^t \mathbf{d_x} + \mathbf{r}^t \mathbf{d_x} \\ \text{subject to}: \quad & \left(\nabla_{\mathbf{x}} \mathbf{h}\left(\mathbf{x}^t\right)\right)^T \mathbf{d_x} + \mathbf{h}\left(\mathbf{x}^t\right) = \mathbf{0} \\ & \left(\nabla_{\mathbf{x}} \mathbf{g}\left(\mathbf{x}^t\right)\right)^T \mathbf{d_x} + \mathbf{g}\left(\mathbf{x}^t\right) \leq \mathbf{0} \end{aligned} \tag{2.14}$$

The matrix B^t must be symmetric and must represent the behaviour of the curvature of the problem, vector $\mathbf{r}^t$ is usually chosen to be the gradient of the Lagrangian at $\mathbf{x}^t$. The choice of symmetric matrix B^t is one of the critical questions of the design of an SQP algorithm, it is generally chosen to be the Hessian of the Lagrangian or an approximation of it [49]. The use of an exact Hessian (if available) leads to fast local convergence thanks to the relationship between SQP and the Newton method. However approximations of the Hessian permit the tailoring of matrix properties to improve far-field convergence of the algorithm.

SQP algorithms have been some of the most effective and widely used gradient based optimisation methods in aerospace applications in the last 15 years. There are significant variations in implementations depending on the method used to compute matrix B^t. SQP coupled with quasi-Newton methods for the approximation of the Hessian is the main optimisation engine of Sparse Non-linear OPTimizer (SNOPT) [50], the open source NLOPT and is implemented in MATLAB's *fmincon*. The most popular implementation amongst the aerodynamic shape optimisation (ASO) community is

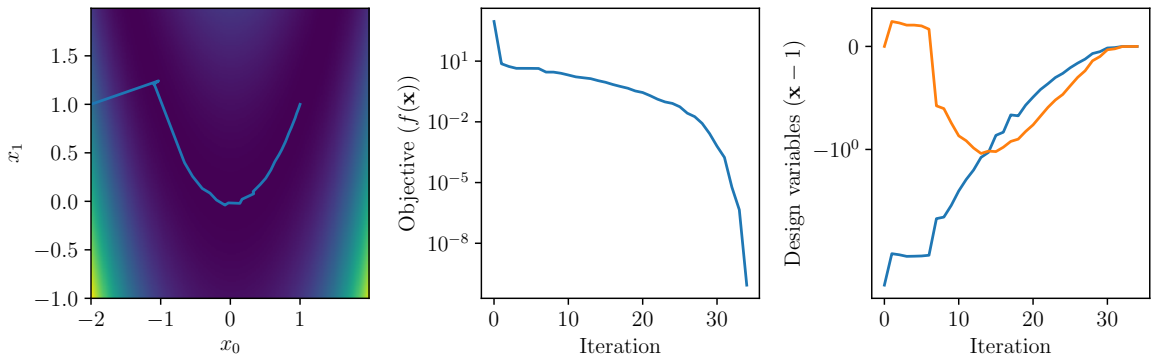


Figure 2.5: Convergence of an L-BFGS SQP on the 2D-Rosenbrock function.

SNOPT which uses a L-BFGS update of the Hessian approximation. Figure 2.5 shows the impressive convergence behaviour of this algorithm on the Rosenbrock function. Research by Papadimitriou and Giannakoglou [51] has shown that the performance of the BFGS update can be significantly improved by using the exact Hessian as a starting point. Subsequently Shi-Dong and Nadarajah [52] have shown that this effect can be achieved with an approximate Hessian evaluated using the adjoint method, making it a cost-effective addition to the traditional BFGS method.

SQP in general, and SNOPT in particular, has been the gold standard in ASO. SNOPT has been successfully used to optimise a wide range of design problems some of which are:

- 2-dimensional aerofoils by Poole et al. [53];
- 3-Dimensional wings and split winglets by Gagnon and Zingg [54];
- structural topological optimisation with pressure loads by Lee and Martins [55];
- aero-structural optimisation of a full aircraft by Kenway and Martins [56].

SQP has also been used in the comparison of parameterisation methods by Master et al. [9, 57] and the study of aerodynamic multi-modality by Chernukhin and Zingg [58].

2.1.2.7 Convergence of gradient based optimisers

So far the discussion of local optimisation has focused on various methods used for optimisation and how they compare to each other. Some discussion of the convergence behaviour has been provided but this raises the question: what does it mean for an optimiser to converge, and how can it be tested? The conditions used to establish optimality of a solution are the eponymous Karush-Kuhn-Tucker condition (KKT) [59, 60]. These conditions can be separated into the first order necessary conditions, and the second

order sufficient conditions.

For the discussion of the KKT condition it is useful to define the Lagrangian ($\mathscr{L}$) of the general NLP defined in Equation 2.1 as an exactly equivalent problem. Dual variables are used in its expression to integrate the constraints into the objective in much the same way that IPMs used barriers and penalties to apply the constraints. These dual variables associated with the constraints are: the *Lagrange multipliers* ($\boldsymbol{\lambda}$) and *slack variables* ($\boldsymbol{\mu}$) for equality and inequality constraints respectively.

$$\min_{\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}} \quad \mathscr{L}(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) \qquad \text{with}: \quad \mathscr{L}(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = f(\mathbf{x}) + \mathbf{h}(\mathbf{x})^T\boldsymbol{\lambda} + \mathbf{g}(\mathbf{x})^T\boldsymbol{\mu} \tag{2.15}$$

The first order conditions are easily understood as the need for the optimum to be a stationary point (a point where the gradient of the optimisation problem is 0) and that the program is feasible at the optimum (the constraints need to be satisfied). This is formally expressed for the minimisation problem expressed in 2.1 in the equation below (Equation 2.16). Starred (*) quantities denote a value at an optimum point.

$$\begin{aligned}
\text{Stationarity:} \quad & -\nabla_{\mathbf{x}} f(\mathbf{x}^*) = \nabla_{\mathbf{x}}\mathbf{g}(\mathbf{x}^*)^T\boldsymbol{\mu}^* + \nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x}^*)^T\boldsymbol{\lambda}^* \\
\text{Primal feasibility:} \quad & \mathbf{g}(\mathbf{x}^*) \geq \mathbf{0} \quad \text{and} \quad \mathbf{h}(\mathbf{x}^*) = \mathbf{0} \\
\text{Dual feasibility:} \quad & \boldsymbol{\mu}^* \geq \mathbf{0} \\
\text{Complementary slackness:} \quad & \mathbf{g}(\mathbf{x}^*)^T\boldsymbol{\mu}^* = \mathbf{0}
\end{aligned} \tag{2.16}$$

To guarantee convergence on non-linear problems the additional second order sufficient condition is required. These are introduced to guarantee that the stationary, feasible solution to the *necessary conditions* curves towards higher values in all feasible directions. This condition checks that a minimum has been found and not a saddle point; it is formally expressed in Equation 2.17

$$\begin{aligned}
&\mathbf{s}^T H_{\mathbf{x}} \mathscr{L}(\mathbf{x}^*,\boldsymbol{\lambda}^*,\boldsymbol{\mu}^*)\mathbf{s} \geq 0 \\
&\text{with}: \quad \mathbf{s} \neq \mathbf{0} \quad \text{and} \quad \left[\nabla_{\mathbf{x}}\mathbf{h}(\mathbf{x}^*),\ \nabla_{\mathbf{x}}\mathbf{g}(\mathbf{x}^*)\right]^T \mathbf{s} = \mathbf{0}
\end{aligned} \tag{2.17}$$

Figures 2.6, 2.7 and 2.8 show the convergence behaviour for the optimisers presented in this section on the 2-dimensional and 5-dimensional Rosenbrock function (Appendix A.1). As expected, the Nelder-Mead algorithm, which does not use gradients, has a very poor initial convergence in the higher dimensional design space. The conjugate gradient shows its limitation in terms of neighbourhood convergence: it is dramatically outperformed by the BFGS and SQP algorithms; which, like the conjugate gradient

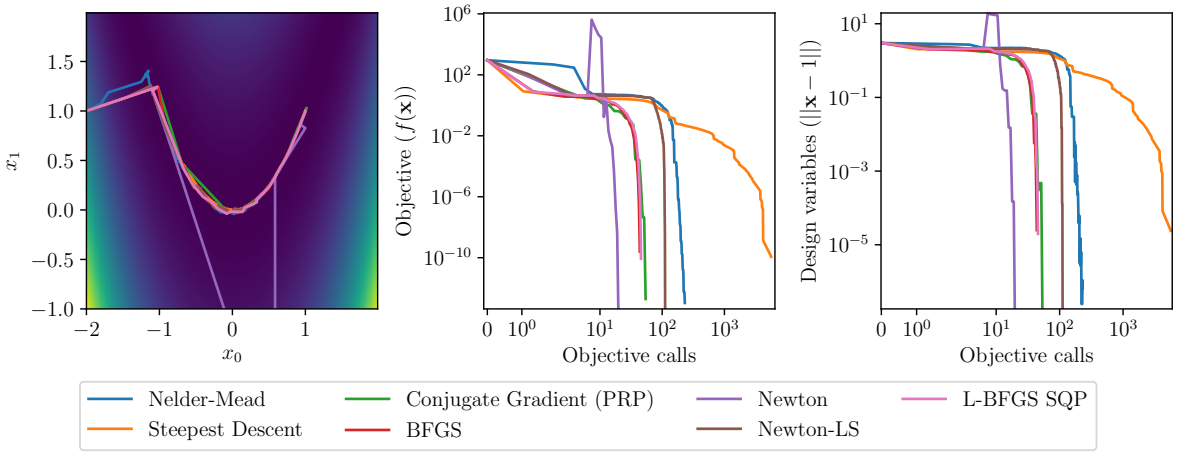


Figure 2.6: Summary of the convergence of the local optimisers on the 2D-Rosenbrock function, the minimum is at (1, 1).

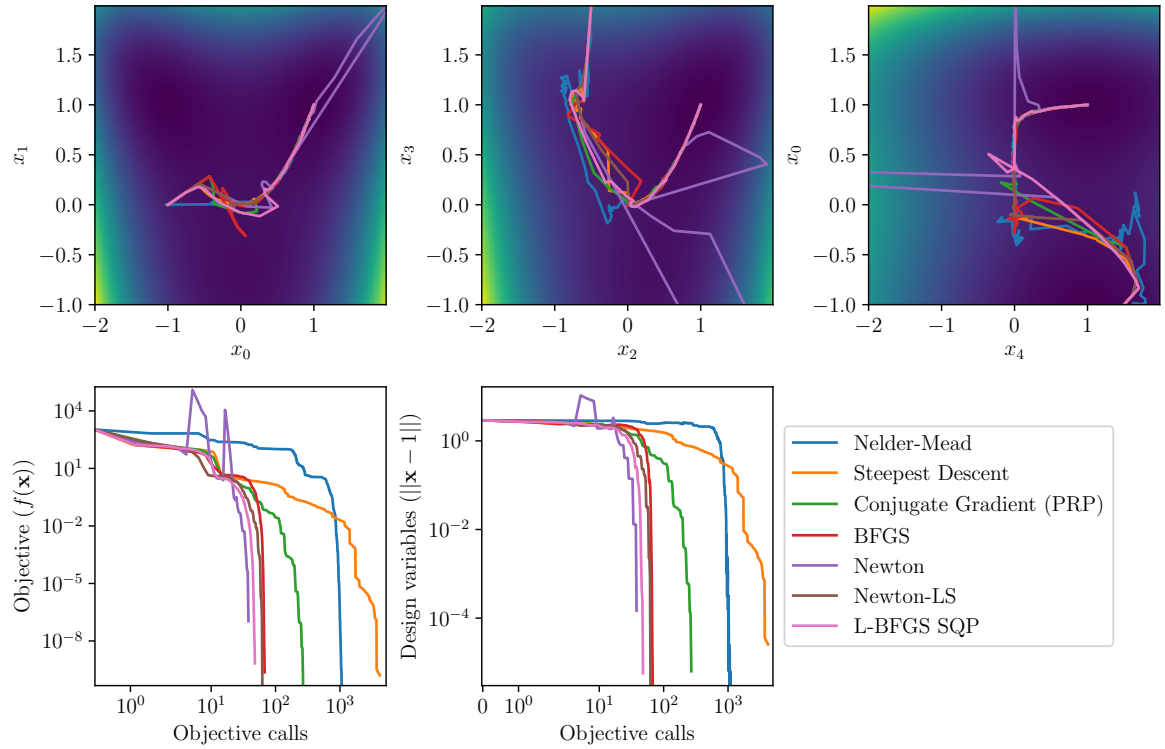


Figure 2.7: Summary of the convergence of the local optimisers on the 5D-Rosenbrock function, the global minimum is at **1** (vector of all ones).

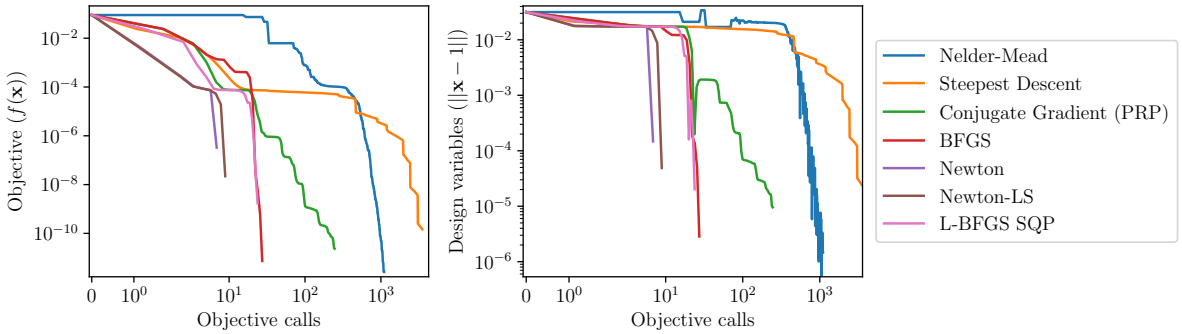


Figure 2.8: Convergence of the local optimisers on the 10D-Rosenbrock function for $\mathbf{x_0} = \mathbf{0.95}$, the global minimum is at $\mathbf{1}$.

method, only use first derivatives. However far from the optimum the conjugate gradient (CG) has very comparable performance to the Newton and SQP algorithm. Additional convergence data associated with these plots is available in Appendix A.1.

2.1.3 Global Optimisers

All optimisation methods presented up to this point have been designed to efficiently find a local minimum; they do not include mechanisms to explore regions of the design space beyond the current local minimum. The algorithms discussed in Section 2.1.2 all require the problem to be continuous, at the very least, and often differentiable. Many problems in economics, mathematics and engineering do not fit this definition; other methods are necessary. To fill this gap in the capability of gradient based optimisers a range of stochastic algorithms capable of exploring multi-dimensional non-convex design spaces have been developed. Unlike the previous methods which were rooted in formal mathematical programming, these algorithms are metaheuristics: they are defined by a set of rules which have been found to be effective experimentally. In general, successful algorithms for global optimisation have two traits: they are agent based and non-deterministic. All of the metaheuristics used for global optimisation contain random processes to help with domain exploration, allowing them to explore beyond a local minimum. Unlike local optimisers, agent-based methods keep a number of simultaneous solutions to the problem, marching all of them according to heuristic rules every iteration. The heuristic nature of these algorithms means that they are designed to find 'good enough' solutions to a problem rather than provably optimal ones.

The loose definition and requirements for the definition of a new metaheuristic has led to the development of tens of algorithms; performance of which is often contingent

on (sometimes) unpublished settings. Selection of an algorithm for an application is dependent on the smoothness of the objective and constraints; as well as the processes that build up the algorithm: its features (swarm, gene, randomness, smoothness) need to be meaningful in the context of the optimisation problem. For this reason a review of all global search methods is beyond this work: only the most established methods and those which have seen applications in ASO are considered. Especially of interest are methods which can handle continuous type problems, with a moderate number of design variables as this tends to be the type of problems explored in ASO. The two main categories discussed here are swarm algorithms (sec 2.1.3.1), which includes particle swarm optimiser (PSO), and evolutionary algorithms (EAs) such as genetic algorithm (GA) (Section 2.1.3.2).

2.1.3.1 Swarm algorithms

In nature swarms are capable of rapidly exploring large expanses in search of food. While an individual taken alone would be incapable of achieving its goal, the swarm and the ability of its members to communicate allows it to achieve its goals. Recognising this, swarm algorithms mimic this behaviour, achieving exploration of the design space by using populations of agents guided by simple rules. The most famous and successful of these algorithms is particle swarm optimiser (PSO) developed by Kennedy and Eberhart [61]. PSO uses a population of particles whose individuals movement is controlled by a mixture of momentum and attraction to the best member of the population. This simple heuristic, when combined into a population, leads to intricate and effective search patterns. The simplicity of PSO and its embarrassingly parallel nature have made it a very popular algorithm in research and industry. It has seen some use in aerodynamics for aerofoil optimisation [62] and for configuration optimisation [63].

Other notable swarm methods are ant colony optimisation (ACO) and the gravitational search algorithm (GSA). The ACO is based on the foraging behaviour of ants and is especially powerful in combinatorial problems such as routing and scheduling problems [64]. GSA is a promising metaheuristic based on a Newtonian gravity model with particle masses related to the value of the objective. GSA has been compared to SQP in aerofoil optimisation [53] and was found to be competitive: while requiring many more flow solutions it was able to find a lower minimum in the 2-dimensional cases considered.

Many other swarm algorithms have been developed to replicate a range of biological behaviours; these include the Krill herd algorithm [65], Grey Wolf Optimizer [66], Wolf Pack Algorithm [67], Artificial Immune Systems and the Firefly Algorithm [68]. The

recent proliferation of heuristic methods based on emergent animal behaviours makes the choice of an algorithm difficult, especially as benchmarks are rarely the same between papers and convergence properties are not systematically studied. Considered in terms of the "no free lunch" (NFL) theorem[1] [69]; this diversity of specialised algorithms is unlikely to help answer meaningful academic questions or industrial needs. The limited amount of research performed on each of these exotic algorithms leaves questions of parameter tuning and theoretical properties unanswered.

2.1.3.2 Evolutionary algorithms

Evolutionary algorithms are also a popular type of metaheuristic which has seen application in industry and academia. They take their inspiration from natural selection and rely on the combination of members and the introduction of mutations to generate new candidates for the population. The most famous of these methods are genetic algorithms (GAs) which were developed by John Holland in the 1960s to mimic Darwinian evolution. The GA is initialised from a population with sufficient variability; then the iteration proceeds in four steps: selection, crossover, mutation and removal. The iteration process is repeated until a termination condition is reached: it can be based on maximum fitness, average fitness, population stagnation or iteration budget. Like swarm algorithms, GAs are embarrassingly parallel and simple to implement; moreover, they can be adapted to work well with discrete problems.

While very effective on a large number of problems GAs suffer from the large number of tuning parameters. Holst and Pulliam [70] provide one of the best available studies of all parameters and go some way to defining some recommendations, however these remain dependent on the modality of the objective function. This observation pulls at one of the main limitations of global optimisers; their effective use relies on an intimate understanding of: the algorithm, the main parameters, and, properties of the optimisation problem that is being tackled. In addition, most algorithms have many variants which have specific strengths, weaknesses and implementation details; making the selection of the most effective one for a given problem difficult.

Differential evolution (DE) is a family of optimisers which has been shown to perform well compared to other stochastic algorithms on continuous style problems [71, 72]. Originally developed by Storn and Price [73], it has since been used successfully in electrical engineering, chemical engineering and data science [74]. These method have seen rapid

[1]The "no free lunch" theorems state *"that if an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems" [69].*

adoption in the optimisation community for constrained non-linear optimisation [75]. DE algorithms, unlike other EAs, have only three tunable parameters; this makes getting the best performance from these algorithms on a new problem simpler. Thanks to the limited number of parameters, a number of self-tuning strategies have been designed [76], streamlining the effective use of DE on new problems.

While DE has few parameters a large number of variants have been devised, altering the difference rules and the selection of members used in the mutation operator. The issue of algorithm selection is amplified when considering constraint handling methods for global optimisation with seven distinct approaches being developed and used with various search algorithms [75]. $DE/rand/1$ was chosen in 2016 as the algorithm used to perform global design space search in this work. Since that decision, new research by Poole et al. [77] has shown that other variants would have been more effective. Despite these findings, the DE variant was not changed; this is to facilitate cross-comparison of later results with earlier ones without needing extensive re-runs at unacceptable computational expense.

GAs have been used extensively in engineering applications to solve problems including structural topological optimisation [11, 78], Aerodynamic Shape Optimisation [32, 58, 79] and Electromagnetic design. While GAs are very capable of performing a global search, they lack the rapid convergence behaviour of gradient based optimisation close to a minimum, leading to criticism in structural topology optimisation (STO) where gradient based methods are very effective [80]. This has usually limited their use to cases with very few design variables, low fidelity solutions or in investigations of the modality of a problem [58, 81, 82].

2.1.3.3 Hybrid methods

For all the methods presented so far there is a clear trade-off between number of objective function evaluations and the thoroughness of the global search. In order for efficient global search on expensive objective functions to be possible, the natural evolution of the current methods is to integrate gradient based and global optimisers into a single search method. Evaluation of local search algorithms is simple: the faster the convergence, the better. For global search metaheuristics a fast convergence of the population can be detrimental, slow convergence is required to provide sufficient exploration of the design space. This has led to a number of approaches coupling GAs and local search methods, leveraging the exploratory behaviour of GAs and the convergence properties of local search methods.

Foster and Dulikravich [32] experimented with the integration of a Genetic Algorithm and the Nelder-Mead simplex method, they were successful in using their method on hypersonic cases. By far the most interesting work on the hybrid optimisers in ASO was carried out by Chernukhin and Zingg [58]. Their work compares the performance of a Multi-Start SQP (MS-SQP), a standard GA and an hybrid GA using some SQP steps to converge locally (GA-SQP). Their results showed excellent performance for the hybrid method on multi-modal design spaces. However their investigation of aerofoil optimisation and configuration optimisation showed that these aerodynamic problems are only weakly multi-modal, which meant the GA-SQP algorithm did not outperform MS-SQP.

2.1.4 Considerations in the Choice of Optimisation Algorithms

In engineering applications choosing between a global and local search algorithm can be hard: the behaviour of the objective function is often not well understood, high dimensionality might limit visualisation options and the computational cost of a single objective call may be high. The selection of an optimisation method is not trivial as it will affect both the time for a design to be generated and its performance. To choose an effective algorithm for a given process two main properties need to be considered: convergence, and, robustness.

2.1.4.1 Optimiser convergence

Theoretical and empirical convergence properties are extremely important to real world applications of optimisation methods: they govern the computational expense and the trust-worthiness of a result. The problem of convergence is very different for gradient based methods and stochastic processes. For gradient based optimisers the study of optimality is made possible by the existence of the *necessary* and *sufficient* KKT optimality conditions discussed in Section 2.1.2.7. Analytical convergence of gradient methods is quantified analytically for most methods [41, 45, 46]. The availability of these proofs provides the analytical backing and the confidence for the use of Quasi-Newton and SQP algorithms on computationally expensive problems where the answer is unknown.

The situation for global search methods is different: there are no generalised methods and processes for guaranteeing the local or global convergence and optimality of an algorithm. The first difficulty with the analysis of heuristic optimisation methods is the lack of theoretical guidance from the outset. For some of the established metaheuristics

theoretical work to establish convergence has been performed (EAs [83], GAs [84], ACO [85], PSO [86] and DE [87, 88]), but these predicted behaviours do not always extend empirically, especially for large scale applications. The difficulty in tuning an individual algorithm means that there is a limited number of systematic comparative studies where each algorithm is tuned at its best performance [75]. This drawback affects the ability of users of these algorithms to choose the best available methods.

2.1.4.2 Sensitivity and robustness

In industrial applications it is not enough to consider an optimal solution on its own: the effect of small disturbances to the problem set-up on the optimal objective function value yields important information about the case being studied. This analysis of the *sensitivity* of the profile is well established for gradient based methods [89, 90]: it gives an analytical expression for the response of design variables and objective to changes in the definition of a NLP.

The stochastic nature of agent based optimisation means that analytical sensitivity analyses are not possible and must be carried out explicitly. In practice sensitivity analysis tend to be replaced by studies of the variability of the solution with repeated runs or alternate optimiser settings. While not a sensitivity analysis in the pure sense; the work carried out by Holst and Pulliam [70] quantified the effect of parameters on a GA algorithm.

Sensitivity is closely related to the concept of robust design which aims to produce optimal designs which are tolerant to small variations in design variables. The idea of robustness as a minimisation of variability (or sensitivity) was pioneered by Genichi Taguchi [91] for quality control. Robust design enhances the original objective function with a measure of the sensitivity, this means the optimal solution is not necessarily the minimum of the objective function but the best compromise between minimising the objective and maintaining low sensitivity. The review of Park et al. [92] presents the main analysis methods for robust optimisation for objective and constraint functions. A significant body of work in ASO considers the robustness of optimisation solutions for 2-dimensional cases [93, 94].

Robustness is a big issue in design optimisation and has been a significant concern of the ASO community over the last few years. A number of benchmark cases (discussed in Section 2.4.3) have been shown to be degenerate when considering optimisation at a single condition. Evidence suggests that reformulating objective functions in terms of a number of simultaneous design points will achieve a similar effect to “robust” optimisers

with much less need for the development of new tools [95].

2.1.5 Optimisation for Design Space Exploration

While a local optimiser may be able to achieve significant improvements on hundreds of thousands of design variables (Aage et al. [14]) that optimised design may be compromised by poor off design performance. Global optimisation is a challenge in many dimensional space, and achieving sufficient design space exploration is a challenge in itself. For this reason methods that use optimisation specifically for design space exploration and mapping have been developed.

Instead of looking exclusively for a strict optima of a specific problem, these methods explore the design space identifying all local optima, or identifying families of solutions. These methods can be very powerful early in a design process to suggest to a designer what direction a design might be taken into. This type of approach can facilitate the set-up of an optimisation: indeed, optimisers are most effective at finding the flaws in the cases they are presented with rather than finding genuine, practical optima. This section will address methods which are designed to gather knowledge about a design space beyond strict minimisation. It includes discussion of niching, multi-objective optimisation, data driven approaches and Illumination methods.

2.1.5.1 Global optimisers for niching

The reason for using global optimisers is the suspicion that a case under consideration is a multi-modal optimisation (MMO) problem. The most natural extension of global optimisation into the idea of full design space exploration is the attempt to discover all the minima of a function. These methods allow designs to be identified which may be close in objective function value but separated in the design space. Optimisation is used in this context to identify regions of interest in the design space which then get refined by an expert [96]. This type of approach works around the need to express all constraints mathematically which can be extremely challenging.

While the idea of niching dates to the early 1970s, it has only become a realistic option as the amount of computing power and the capability of agent based optimisers have improved [96]. These advances have seen the use of niching algorithms for drug molecule design, scheduling problems and space mission design [96]. In this way optimisation can be used at an earlier design stage highlighting features of highly dimensional and

discontinuous design spaces. A recent study by Poole and Allen [77] compare DE niching algorithms opening up the opportunity to efficiently tackle practical constrained MMO.

2.1.5.2 Multi-objective optimisation

Many real world optimisation problems involve a trade-off between different objectives and as such are not readily formulated into the traditional NLP formulation (Equation 2.1). In optimisation, these trade-offs are expressed as the minimisation of two distinct, competing, objective functions with the same design variables (Equation 2.18). This type of trade-offs is known as a multi-objective optimisation problem (MOOP); two main approaches exist to tackle them: reduction to a single objective, or direct methods [97].

$$\begin{aligned} &\min_{\mathbf{x}} f_1(\mathbf{x}) \quad \text{and} \quad \min_{\mathbf{x}} f_2(\mathbf{x}) \\ \text{subject to}: \quad &\mathbf{g}(\mathbf{x}) \geq \mathbf{0} \\ &\mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned} \tag{2.18}$$

Methods to tackle MOOP need to resolve how to interpret the “and” between the objectives in 2.18. In some cases objective functions can be combined into a single broader objective in a meaningful way. This can be the case in economic or engineering problems where the final objective of a design can be captured rather than component level objectives on which optimisation is usually carried out. An example of this would be the design of the structures and aerodynamics of an aircraft being traded-off in terms of its operating cost. However there are as many cases where these meaningful relations do not exist or are too complex to establish mathematically. In these cases a single objective function can be artificially composed in the form of $f = w_1 f_1 + w_2 f_2$ where weights w_1 and w_2 balance the relative importance of the objectives. This allows the use of traditional single-objective optimisers.

Often, when those indeterminate multi-objective cases are encountered, a designer is interested in the behaviour of the optimal designs for a range of weights. Optimal solutions for given weights are called *undominated* or *Pareto optimal* solutions; the set of all such solutions is called the Pareto front of a MOOP. A naive approach to building this front would be to repeat optimisations with a range of weights; however, in the same way that niching algorithms recover all the minima of a function in a single optimisation, bespoke algorithms can achieve the same for MOOP [97]. Some of these methods are closely related to constraint handling methods for global optimisers [75]; this makes

intuitive sense as constrained optimisation can be understood as the optimisation of the minimisation of an objective and of infeasibility.

Multi-objective approaches are best used to tackle problems where the trade-off values are not well known or likely to change. An example of this type of problem is the design of an electrical grid including batteries, wind, solar and diesel systems by Dufo-Lopez et al [98]. In this work, life cycle emissions and cost of emissions are optimized using a MOOP strategy allowing to establish optimal approaches depending on different inflation scenarios. MOOP approaches go beyond traditional engineering and can provide guidance in policy decision making.

2.1.5.3 Model driven approaches

Optimisation methods perform calls of an objective function with the goal of finding a well performing solution or a set of them. Agent based optimisers, niching and multi-objective optimisation methods aim to discover more optima by increasing the exploration of sub-optimal designs within the design space. An alternate approach is to explicitly design algorithms for design space exploration and modelling of the response of the objective function and then look for points of interest on that modelled response of the objective. This type of approaches are known as *surrogate modelling* methods.

Surrogate modelling methods are often used to replace computationally expensive objectives or constraint with a function approximation which is much cheaper to evaluate. The surrogate is usually built from a limited number of data points; from which it allows an estimation of function values at other locations in the design space. The general formulation of a surrogate function $s(\mathbf{w}, \mathbf{x})$ to an objective $f(\mathbf{x})$ is presented in Equation 2.19. In this equation a known parametric formula $s(\mathbf{w}, \mathbf{x})$ is tuned to match data using $\mathbf{w}$ parameters. Surrogate models are a very broad family of methods ranging from polynomial regression and response surface methods to ANNs used for artificial intelligence.

$$\begin{aligned}
\text{For}: \quad & f(\mathbf{x}) \text{ defined for } \mathbf{x} \in \Omega \\
\text{From a set of samples}: \quad & \left\{ f(\mathbf{x}_0), \cdots, f(\mathbf{x}_i), \cdots, f(\mathbf{x}_n) \right\} \\
\text{define}: \quad & s(\mathbf{w}, \mathbf{x}) \quad \forall \mathbf{x} \in \Omega \\
\text{with tuning} & \\
\text{parameters } \mathbf{w}: \quad & \min_{\mathbf{w}} \quad \sum_{i=0}^{n} e(\mathbf{w}, \mathbf{x}_i) \quad \text{where} \quad e(\mathbf{w}, \mathbf{x}) = |f(\mathbf{x}) - s(\mathbf{w}, \mathbf{x})| \\
\text{such that} &
\end{aligned} \tag{2.19}$$

The most common surrogate modelling methods at the moment are ANN which combined with Machine Learning (ML) are ubiquitous throughout a range of industries [99]. The very flexible formulation and scaling of the networks makes them a very powerful tool for extracting correlations and insights from arbitrary databases. In ML the tuning stage of the network is done explicitly as an optimisation using a method called back-propagation [23] combined with stochastic gradient descent.

For continuous functions the most widely used method is the Kriging interpolant. Originally developed in the 60s and 70s by Matheron [100, 101] for mapping of geographic data around the available surveyed points, it aims to minimise the prediction error of a model based on statistical analysis. Kriging methods are interpolants which means that for all samples $\mathbf{x}_i$ the error e is equal to zero. Kriging models are a linear combination of basis functions with a one to one equivalence in the number of bases to the training samples. This relationship means that the Kriging response surface can always interpolate the training dataset regardless of its size. These methods and the similar radial basis function (RBF) interpolation method have successfully been used to interpolate between aerodynamic properties of an aircraft [102, 103].

While very effective in problems with low dimensionality these methods struggle to map responses in higher dimensions due to the cost of sufficiently sampling the design space. Accurate representation of the trends in each dimension requires an exponential amount of samples which for expensive objective functions rapidly becomes impractical. While efficient approaches exist to sample design spaces, there is no going around the "curse of dimensionality" [104]. This limitation has been the Achilles heel of surrogate-based optimisation methods in aerodynamics, which rely on very large numbers of samples to achieve results comparable to local optimisers.

While the use of monolithic surrogate models for optimisation may not always be effective, hybrid approaches can yield good results. The method by Ong et al. [105] uses a local surrogate to reduce the computational cost of a GA. One relatively recent approach to tackle the dimensionality problem is the use of active subspaces of a function. The culmination of these approaches was performed by Li et al. [106] in which a surrogate model of airfoil viscous compressible flow behaviours was generated on a design space defined by singular value decomposition (SVD) modes extracted from airfoil databases. This surrogate model allowed optimisation of airfoils in Reynolds-Averaged Navier-Stokes (RANS) flow with 0.04 to 2.5 drag counts[2] of accuracy in a few seconds, potentially allowing optimal interactive design. In these methods, a small subset of the

[2] 1 drag count is equivalent to a drag coefficuient (C_D) of 0.0001.

most important dimensions of a function are identified in a given region and a surrogate is built for this reduced design space [104]. These approaches are closely related to dimensionality reduction methods such as proper orthogonal decomposition (POD) and SVD [107, 108] which have been used to re-parameterise design spaces. This idea of data driven re-parameterisation is useful and highlights how the separation between parameterisation and optimisation method is blurred by the development of surrogate modelling methods.

2.1.5.4 Quality diversity through illumination algorithms

The final methods discussed in this section are known as quality diversity (QD) algorithms [109] and depart from all other optimisation methods discussed up to this point in that they are expressly interested in sub-optimal designs. Originally pioneered in robotics and artificial intelligence (AI), these algorithms explicitly look to build maps of the design space or constraint space where well-performing designs are identified within each region of the design space. These algorithms have been used to let robots adapt very rapidly to damage and automatically reconfigure their control laws [110].

Instead of finding a single optimum, illumination algorithms find the best solutions in every region of a feature space. The feature space is a set of properties of a design which are of interest to a designer; for a physical system this could be the weight, the height and the curvature. This feature space is segmented in a grid pattern and the algorithm identifies the optimum in each region. The philosophy of this type of algorithm is similar to a combination of multi-objective optimisation and surrogate modelling: the goal is to map the optimal solutions in the feature space. Once the map has been built a change of requirements new good designs can be chosen from the map without re-runs of optimisation methods.

Illumination methods have been used in aerodynamic design optimisation by Gaier et al. [111, 112]. By integrating illumination with surrogate methods, they were capable of performing the conceptual design of “velomobile” tricycle vehicles designed for human powered speed and distance records. In this initial design study the surrogate assisted illumination algorithm successfully highlighted exotic and effective designs.

2.1.6 Design Optimisation

The specific branch of optimisation with which we are concerned in this thesis are “design optimisation” cases. In design optimisation the decision variables of the optimisation

are translated into a two dimensional or three dimensional geometry; with the aim of producing a manufacturable and workable geometry. The challenge of design optimisation methods is to convert the limits and properties of an engineering task into a series of tractable optimisation problems.

For this to be achieved design optimisation requires the integration of optimisation or exploration methods, with a physical analysis method. In this context the physical analysis method becomes the objective function of the optimisation; and the design variables control the geometry. The translation of the optimisation variables into a physical design is permitted by a *parameterisation method*. The parameterisation governs how easily the optimisation problem will be navigable by the optimiser and has a large impact on the properties of the optimisation problem. Parameterisation methods are bespoke to their application and those specific to aerodynamics will be discussed in later Sections (2.4.2). One of the key challenges of "design optimisation" is to avoid the trivial answers that will arise in the naive formulation of a design problem. There are two main approaches to these problems: constraint formulation and multi-disciplinary optimisation.

2.1.6.1 Avoiding trivial and degenerate answers in design problems

The natural formulation of many design problems appears very simple: minimise the weight of a structure, minimise the drag of an aerofoil, maximise the thrust of an engine. Behind these formulations lies a very large number of implied constraints which make these design problems difficult to tackle with optimisation. An optimiser tackling the drag minimisation of an aerofoil will rapidly reduce the geometry to a non-lifting flat plate; while correct this is a trivial result.

Taking the case study of the optimisation of an aerofoil; to achieve a meaningful result, some physical constraints must be added, relating to the wing from which it is extracted. These constraints must reflect the competing requirements of structural supports necessary to practical lift generation. These ideas need to be formulated into the optimisation problem by way of constraints. A minimal lift constraint can be envisaged but this represents a significant increase in the complexity of the optimisation problem: it is now an unequally constrained fully non-linear optimisation problem. This constraint can be simplified to an equality constraint that is satisfied at every step by varying the angle of attack. This tuning at every step would increase the cost of the flow solve, alternatively the angle of attack can be included as a decision variable for the optimiser to use [113]. Both approaches have been used in literature and come to illustrate some

of the challenges when framing engineering cases as optimisation problems.

The second constraint required for aerofoil optimisation is the representation of structural limits of a wing. The challenge is that the structural constraint is very application dependent, the structures of a helicopter blade will be very different to the wing of an airliner. This means that at early design stages and in academia less specific constraints which are cheap to evaluate are often desirable. It is common for aerofoil optimisation to use area constraints as a surrogate for the structural requirements of an aerodynamic body. This approach has limitations as it implies that each discipline optimises with limited input from other important design considerations. These intrinsic limits to discipline based design optimisation can be overcome by using multi-disciplinary optimisation approaches.

2.1.6.2 Multi-disciplinary optimisation

Multi-disciplinary optimisation (MDO) methods integrate the disparate discipline specific modelling and optimisation methods into a single framework where many more of the interactions and feedback loops are accounted for, not just through a few constraints. MDO methods are popular in aerospace applications as the systems are very tightly coupled requiring deep integration between aerodynamics, structures, propulsion, control and performance calculations. This integration was first formalised by Kroo et al. [114] in an effort to streamline the development and parallelisation of the methods used within this architecture. The process of "collaborative optimization" used in [114] allows parts of the optimisation to be run in parallel by using somewhat out of date information from previous subsystems; but feasibility of the final design is always guaranteed by the formulation of the constraints. This system was successfully applied to the conceptual design of an aircraft. Similar frameworks showed that the aero-structural optimisation of a business jet allowed for better performance than sequential system optimisations [115].

One of the challenges with multi-disciplinary optimisation is that comparison of frameworks and results can be difficult [116]. In their review, Martins and Lambe [116], develop a new formalism to improve the communication of framework structure in the hope of improving re-implementation and reproducibility. Recent work by Brooks et al. [117] aim to provide some element of standardisation through benchmark aero-structural design and optimisation results. Modern frameworks for MDO such as the open source openMDAO framework [118] open up the application of MDO to more cases and users, without variations in implementations.

The second challenge of MDO frameworks is computational cost: combined cost of

the systems can be very high, especially as the number of feedback loops increases the required number of function evaluations. This usually puts agent based optimisation out of the question and requires efficient differentiation methods to drive the update of gradient-based optimisers. [119] For this purpose a differentiation methods cheaper than finite difference methods are required. Martins and Hwang [119] review approaches for differentiation which provide efficient alternatives to finite differencing.

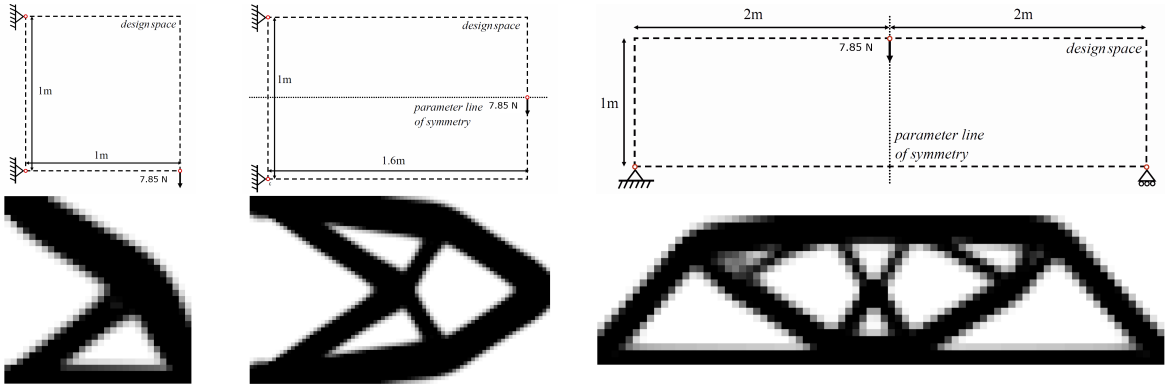
2.2 Structural Topology Optimization

Structural topology optimisation (STO) has been a very active field of research and has seen some industrial application, with the inclusion in established computational aided design (CAD) and structural simulation packages. Deaton and Grandhi [11] provide a complete review of STO; presented in this section is an overview of the different methods available. The goal is to identify features, methods and problems that are likely to be relevant to aerodynamic cases.

Through the last 30 years, STO has seen the development of three principal methods: the solid isotropic material with penalisation (SIMP) method, the evolutionary structural optimisation (ESO) method and the level-set method (LSM). The following sections identify how the topological problem has been tackled in STO: how is the parametrisation able to transition between topologies? How does topology affect the modality of the design problem? And how can changing topologies be integrated efficiently into an optimisation framework?

2.2.1 Density Based Methods: SIMP and Homogenization Methods

Historically, the first methods developed for STO were *homogenization methods*, these rely on the segmentation of the design domain into squares in which the density of a material can be varied, changing part weight and load carrying ability. Early work on this class of methods focused on proving the validity of the relationship between density and stiffness [120, 121]. Homogenization results are used to define the physical properties of partially full design cells; these allow the optimisation process to proceed smoothly from full to empty domain regions. The results from these works were instrumental in the development of SIMP [122], certainly the most widely used STO procedure (example in Figure 2.9). Intermediate levels while physically meaningful are impractical



(a) Cantilever beam. (b) Symmetric cantilever. (c) MBB beam.

Figure 2.9: Example of SIMP solutions from Taylor et al. [124] generated using the 99 line STO code by Sigmund [125].

from an engineering perspective. This is resolved by penalising the intermediate levels of density in the optimisation process. The penalised problem is then solved using a gradient based optimisation method. Density distribution methods have been adapted to 3-dimensions [123].

The other widespread density methods are ESO [126] and bi-directional ESO (BESO) [127]. These methods rest on a heuristic process rather than mathematical programming: material is removed where the internal loading is low. BESO adds the possibility of adding material if it is needed. The optimality of the ESO heuristic has been questioned by Rozvany [13] despite efforts to understand its theoretical behaviour by Tanskanen [128].

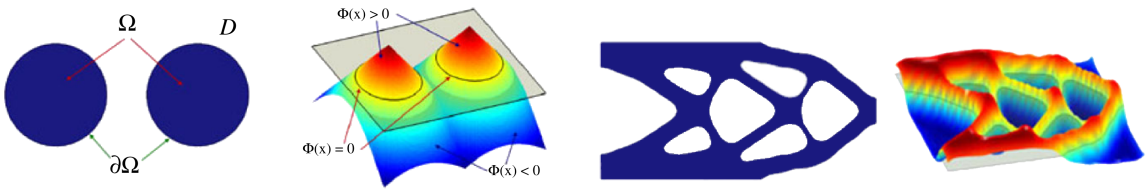
These methods have three main limitations: checker-boarding, multi-modality and directional dependency. Checker-boarding is the formation of unphysical oscillatory patterns within the design space. This is the result of the interaction between parameterisation and structural analysis discretisation. It is generally solved by using filtering methods or higher order finite elements [13]. The starting point for SIMP and ESO is often a uniformly distributed design space, while this makes intuitive sense, the coupling with a local optimisation method is problematic. Effectively the topological optimisation problem is recognised as very multi-modal [129]. The penalty method of SIMP can be coupled with continuation methods (progressive application of penalties) to increase the likelihood of finding the global minimum, however validating that a known global optimum has been found remains difficult [13]. The directionality of the grid also a large influence on the optimum profile; Cartesian grids limits the range of angles that can

be explored by members, artificially biasing the optimisers towards solutions easy to represent on these grids. Finally, the absence of contour representation can limit the range of analysis that can be performed on the profiles generated by these STO methods. Generating a crisp boundary suitable for further analysis or manufacturing from those methods are heuristic processes which may not always capture the intent of the optimiser.

2.2.2 Level Set Methods in STO

The main alternative to homogenization is the level-set method (LSM) introduced by Wang et al. [130] and its subsequent developments. In these methods the structural profile is represented by the level set of a parametric function. These methods were shown to be very competitive and solve the checker-boarding and directional bias of the homogenization methods [130]. The choice of basis function and its support radius for the level-set function (LSF) is important to the performance of the entire optimisation process [131]. Possible choices for the LSF include: finite element method (FEM) Basis functions, RBFs, spectral parameterisation and parameterised geometric shapes. These basis functions are then assembled to form the level-set function (LSF), the structural profile is then extracted as a level set of that function. With an appropriate choice of level-set function (LSF) this yields smooth continuous profiles. Figure 2.10 presents an example LSM. Optimisation using LSFs can rely on three mechanism for changes: boundary profile variations; LSF parameter variations; and topological variations.

LSFs can be controlled using Hamilton-Jacobi (HJ) evolution procedures or mathematical programming methods. The HJ methods use a partial differential equation and a velocity field to drive changes tangential to the structural profile. The profile variations are usually calculated using an adjoint approach [131, 133]. The main difficulty of this



(a) Material domain (Ω) and LSF ($\Phi(x)$) for simple shapes.

(b) Material domain and corresponding LSF generated by the optimisation of a symmetric cantilever.

Figure 2.10: Example of level set material domains and level set functions for an STO problem, from Luo et al. [132].

method is that the desired profile variations must be mapped to the parameterising variables which control the LSF. Mathematical programming methods control directly the parameterisation variables, the preferred algorithms are SQP and the Method of Moving Asymptotes (MMA). Some research has been carried out using global search methods [78], however the high number of design variables causes GAs and other metaheuristics to be prohibitively expensive.

To go some way towards resolving the modality problem introduced by topological changes, the concept of topological derivative developed by Sokolowski and Zochowski [134] is introduced in LSM. This analytical derivative measures the response of the objective function due to the introduction of an infinitesimal hole in the solid domain. This is used to specify where new holes should be introduced. In fluid dynamics the derivation of topological derivatives has been carried out for Stokes flow by Amstutz [135], derivation for the Euler equations could yield some interesting results for ASO.

Interestingly many of the current issues identified by Van Dijk et al. in their review [131] are analogous to those identified for aerodynamic parameterisation: uniqueness of the parameterisation; control of numerical artefacts; avoiding spurious local minima; and smooth control of the geometry. Some of the regularization methods used to improve the LSM performance should be considered in an aerodynamic topology optimisation framework.

2.3 Computational Fluid Dynamics for Optimisation

It is accepted that the behaviour of fluids is entirely described by the Navier-Stokes (NS) [2] equations. Unfortunately these equations are analytically intractable, and proof of the existence or not of analytical solutions are one of the six remaining Clay institute of mathematics “Millennium problems”. As a consequence predicting aerodynamic forces for the design of aerospace vehicles has been a challenge since the Wright brothers first took off. Originally, designers could only rely on wind tunnel experiments, empirical rules and data bases of experiments to guide their effort. Since exact solutions are out of reach approximate solutions to the Navier-Stokes equations can be calculated using analytical simplifications and numerical methods. With the advent of the digital computer, designers turned to the new field of computational fluid dynamics (CFD) to help predict behaviours of aerospace vehicles. In recent years CFD has propagated to many more industries from automotive, civil engineering and biomedical engineering to

video game and animation special effects.

The wide range of potential applications has led to the development of a large number of methods which provide sufficiently accurate results for the given problems. Each approach exhibits a trade-off between physical and computational accuracy, precision and speed; which depends on the assumptions which underlay the model and the quality of any potential discretisation. The discussion in this section focuses on the high level properties of CFD methods used for the design of aerospace systems in view of using them for optimisation since an exploration of the entire field of CFD would be too broad an endeavour. As a consequence, the primary concerns for this section are: what physics are being modelled and can a given CFD method be used in optimisation? Suitability of a method for optimisation will be case and resource dependent but concerns of computational speed, reliability and automation will be taken into account.

2.3.1 Selection of an Approximation to the Navier-Stokes Equations

Computation of a flow solution requires the selection of a numerical or analytical approximation to the NS equation. Numerical solutions of the Navier-Stokes equations can be performed with a cost proportional to $\mathcal{O}(Re^3)$ [136] (where Re denotes the Reynolds Number). Even for flows over simple geometries, direct numerical simulations (DNS) rely on some of the most powerful computers in the world [139, 140](sphere and flat plate respectively). Figure 2.11 shows the most common methods used for external aerodynamics and internal flows, along with their 'lineage' down from the Navier-Stokes equations. The information summarised into this figure is well known and can be found in books, lecture notes and websites [136–138].

Fluid dynamic approximations are derived in three ways: physical assumptions, mathematical reductions, and modelling. Physical assumptions explicitly remove physical phenomena from the equations consequently reducing the size of the system, and its computational cost. Mathematical reductions allow a simplification of the equations without expressly removing a physical phenomena, but making some simplifying assumptions about the behaviour. An example of this is the time-averaging performed in the derivation of the Reynolds-Averaged Navier-Stokes equations: instead of dealing with the time variability of viscous effects they are averaged and then modelled. The final step is to model any behaviour not sufficiently represented by the current approximation; as such model increases the computational cost of the flow solution, but much less than

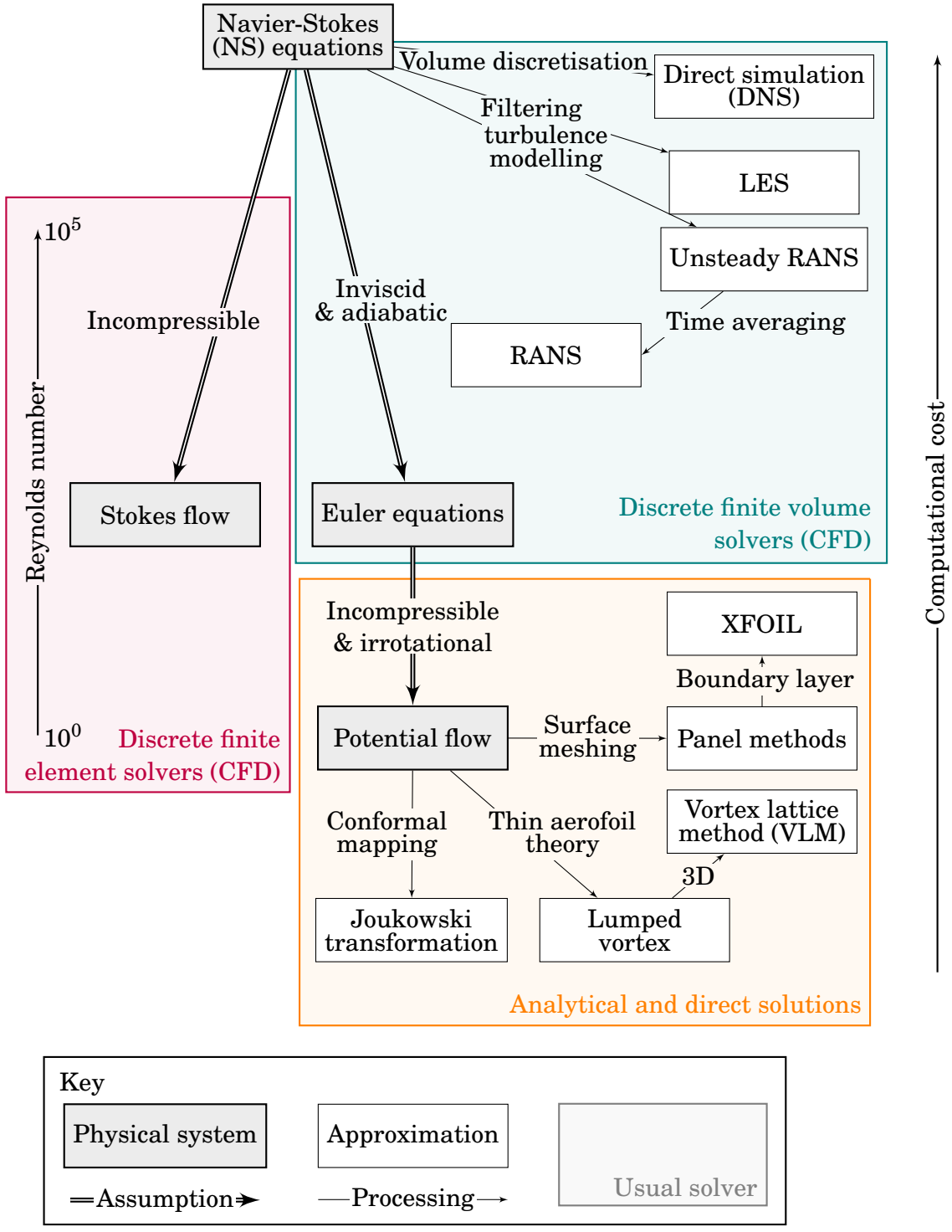


Figure 2.11: Available approximations and solution processes for the Navier-Stokes Equations [136–138]

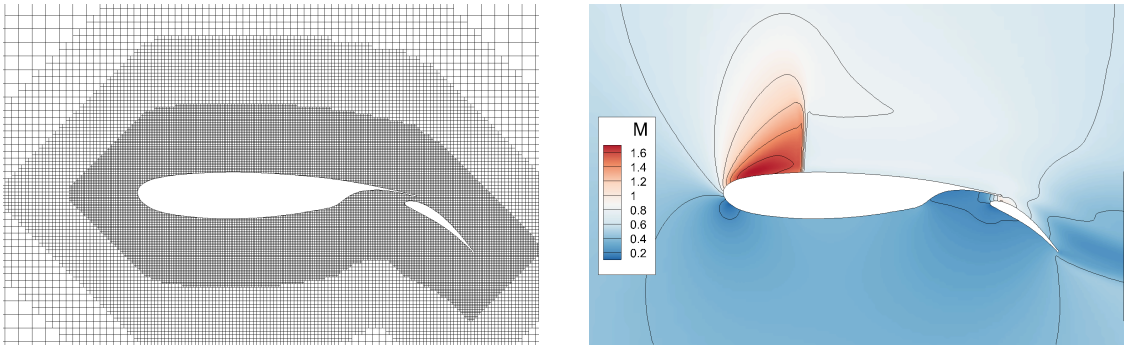
the direct simulation of that behaviour would have cost. Keeping with the example of RANS, the turbulence which is removed by the time averaging is reintroduced in the form of turbulence models which are more readily solved numerically than the original turbulent equations.

When choosing a calculation method, the expected dominating physical phenomena and the range of applicability of any models need to be carefully considered and traded off with the computational cost. Turbulence models are the Achilles heel of RANS: they have narrow validity ranges and are reliant on very specific discretisations to converge [136]; however with careful calibration on experimental data these methods have become the standard for drag prediction in the aerospace industry.

For aerodynamic optimisation and especially for topology optimisation, any selected method must be able to represent arbitrary shapes and produce the correct trends if not the exact drag value. This requirement excludes potential flow methods which rely on the manual specification of stagnation and separation points essentially enforcing a known flow pattern. The other methods in Figure 2.11 are capable of simulating flow dynamics around arbitrary shapes without further assumptions but the discretisation itself is very important to the quality of the result. The cost of unsteady RANS, large eddy simulation (LES) and DNS make their use in optimisation impractical and unnecessary: in most cases either compressibility or viscosity dominate, and consequently Euler or Stokes flow are sufficient. In addition, it is unclear how the unsteadiness could be exploited with current design processes: the results would likely need to be time-averaged in the optimisation process, providing no benefit over RANS. If both compressibility and viscosity are required RANS captures the correct trends between different geometries. In this study optimisation with Stokes, Euler, and RANS approximations is considered; while optimisations with these approximations are routinely performed, they remain extremely expensive and require efficient optimisation methods.

2.3.2 Domain Discretisation

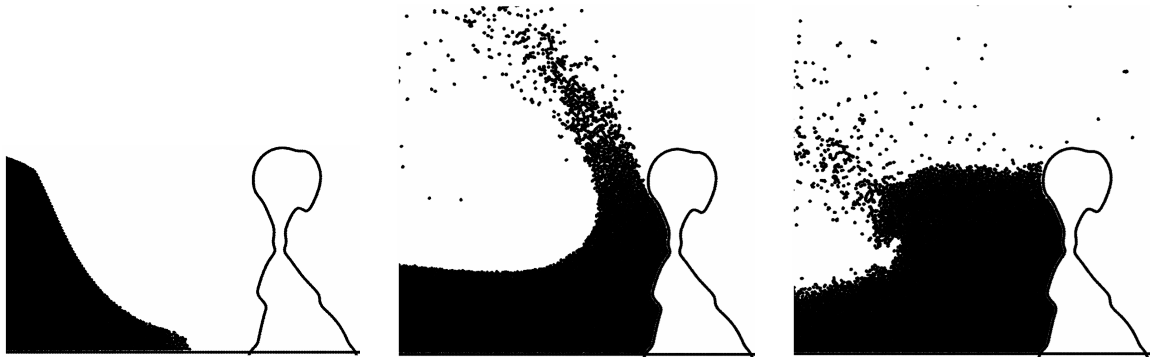
The three equations systems that were selected as candidates for optimisation are numerical methods and as such they require discretisation of the fluid domain. Two main types of discretisation exist: Eulerian formulations where properties are tracked at a location in the flow field (Figure 2.12) and Lagrangian formulations where particles are followed as they traverse the region of interest (Figure 2.13). Eulerian methods are standard in single fluid simulations without free-surfaces as they offer fine geometric resolution and calculation of aerodynamic forces. Conversely smoothed particle hydrody-



(a) Eulerian cut cell mesh.

(b) Mach number flood plots.

Figure 2.12: Transonic flow simulation around a 2 body aerofoil using a cut-cell boundary fitted mesh at a Mach number of 0.5.



(a) Before impact.

(b) During impact.

(c) After impact.

Figure 2.13: Optimised coastal defence using SPH, a Lagrangian fluid dynamics method; images from Hall et al. [141].

namics (SPH), the main Lagrangian formulation for fluid dynamics, is an effective way of computing free surface or multi-fluid flows. The particle in cell (PIC) [142, 143] and the fluid implicit particle (FLIC) [144] methods aim to exploit the benefits of Eulerian and Lagrangian methods but are not yet at the level of accuracy of mesh based methods. For optimisation of external flows the Eulerian approach is standard as most methods require a very precise resolution of the forces to compute gradients to drive gradient based optimisations.

The quality of a flow solution coming from an Eulerian method is highly dependent on the quality of the discretisation of the surface and of the domain. The surface discretisation impacts the resolution of the pressure forces that are integrated to compute global aerodynamic forces such as lift and drag; an insufficient resolution will lead to averaging in the solution and might not capture flow features which impact the global forces.

Once a good surface discretisation is generated the fluid region must also be segmented. Two methods exist: structured and unstructured meshing methods. Structured methods use "blocks" of ordered Cartesian cells; these methods allow the flow solver to access flow properties in neighbours very efficiently leading to significantly faster computation. For streamlined bodies, boundary fitted Cartesian structured meshes have the benefit of having cells aligned with the principal directions of the flow further simplifying analysis of the flow. Meanwhile unstructured meshes allow a wider range of cell shapes and are generally more easily generated around arbitrarily complex geometries.

Desirable properties of the discretisation are dependent on the equations being calculated and the size of the expected flow features. The turbulence models used in RANS simulations are particularly sensitive to the growth rate of cells away from no-slip wall boundaries; requiring very small and smooth cells to accurately model boundary layer evolution.

Unfortunately meshing is a non-trivial task and research into tools facilitating the automatic generation of high quality meshes is ongoing. In optimisation this can be alleviated by generating a single high quality mesh which is then deformed to match the changes of geometry required by the optimiser. The two types of deformation methods used in aerodynamic optimisation are: elasticity models and algebraic methods [145, 146]. However mesh deformation cannot alter the structure of the mesh and as such is not capable of representing topology change.

One approach to resolving the difficulty of automatic meshing is the use of immersed boundary methods (IBMs); this type of method does away with the need to have boundary fitted meshes. IBMs use a Cartesian section of the fluid and solid domain in an Eulerian framework and Lagrangian variables for the solid boundary [147]. IBMs are commonly used for incompressible flow, notably bio-medical flows where boundaries are flexible and the geometries extremely complex [148]. Recent efforts have shown that IBMs can be used to predict the aerodynamic flow features in the compressible regime [149, 150]; however these methods have traditionally not been used for compressible cases where a precise definition of the surface is required to achieve accurate force predictions. Lattice-Boltzmann (LB) methods face similar limitations and are much better suited to the approximation of detached flows. It is only recent developments that have allowed computation of compressible flows at high Reynolds numbers with accurate measurement of incremental changes in the aerodynamic forces [151]. While IBMs and LB methods may not be suitable for compressible cases, their Cartesian octree grids can be used to generate geometry conforming CutCell meshes [152, 153]. These meshes have successfully been

used in inviscid transonic [154] and supersonic [155] optimisations.

2.4 Aerodynamic Optimisation Frameworks

This chapter has explored the various optimisation methods and flow solvers which can be used to guide a designer in the exploration of new design cases. The different properties of optimisation methods and their use in the context of ASO has been considered. This section will look at issues specific to the development of effective aerodynamics optimisation frameworks including: calculation of design derivatives, parameterisation of aerodynamic shapes, and, the standardisation of test cases.

2.4.1 Calculation of Design Derivatives

Due to the high computational cost of CFD methods, gradient based optimisation methods dominate the field of aerodynamic shape optimisation. The ubiquitous use of gradient based optimisers means that efficient and robust methods for computing derivatives with regard to the design variables are required. In ASO, the design variables are some form of geometric property of the profile: these can be the position of individual grid points, global properties (sweep, twist) or prescribed basis functions. There are three methods for computing derivatives with regards to the geometric properties: finite difference; complex step; and adjoint approaches. An overview is presented in the next sections; Martins and Hwang [119] provide an interesting review of the theory and implementation of complex step and adjoint methods.

2.4.1.1 Finite difference

Finite differencing is the simplest method for objective function differentiation. It is based on Taylor expansions to find the derivatives, they can either make use of a forward (or backward) difference or a central difference. These methods can be implemented on any black box system; no knowledge of the behaviour is required as only the design variables and the output are monitored. Each additional evaluations will yield one partial derivative (one column of the Jacobian for vector valued functions), which makes finite differences very expensive for large numbers of design variables.

The equation for the forward difference method and central difference methods are presented below in 2.20 and 2.21 respectively. With $\mathbf{e}_j$ the zero vector for all elements but the i^{th} which is unity and h the step length.

$$\frac{\partial f}{\partial x_j} = \frac{f\left(\mathbf{x}+h\mathbf{e}_j\right)-f(\mathbf{x})}{h}+\mathcal{O}(h) \tag{2.20}$$

$$\frac{\partial f}{\partial x_j} = \frac{f\left(\mathbf{x}+h\mathbf{e}_j\right)-f\left(\mathbf{x}-h\mathbf{e}_j\right)}{2h}+\mathcal{O}\left(h^2\right) \tag{2.21}$$

Finite differences outperform competing differentiation methods in dealing with noisy functions: by taking measurements 'far' apart short wavelength noise can be overcome to obtain a descent direction. However the schemes above are prone to discretisation and numerical errors. This rapidly becomes an issue when using CFD as the precision of f will be dependent on the level of convergence of the flow solver, requiring longer run times for each flow solve to achieve sufficiently accurate gradients. Numerical errors appear in the subtraction process performed by floating point arithmetic: when the step is reduced the disturbed value approaches the reference value which leads to a loss of significant numbers [115]. This implies that there exists an optimum step length where the discretisation error and the floating point error are of the same order, requiring tuning of the finite difference.

2.4.1.2 Complex step

The complex step is an alternative to finite difference methods which does away with floating point arithmetic error induced by the subtraction. The principle is similar to finite differences, the Taylor expansion is used for a step taken in the complex plane [156]. The equation for the derivative using a complex step method is shown below.

$$\frac{\partial f}{\partial x_j} = \frac{\mathrm{Im}\left[f\left(\mathbf{x}+ih\mathbf{e}_j\right)\right]}{h}+\mathcal{O}\left(h^2\right) \tag{2.22}$$

Detail of implementation for CFD is presented by Martins et al. [156]. In contrast to finite differences which require no knowledge of the underlying system, the complex step requires specially adapted code. However existing code can be adapted by using automatic algorithms [115].

By moving the difference to the complex space the numerical error introduced by the subtraction process is removed. This means that the small size of the step does not cause an increasing numerical error: the error in the gradient decreases quadratically with step size and remains at machine precision for very low step sizes ($h < 10^{-15}$) [119].

2.4.1.3 Adjoint solvers

While the complex step process allows for a reduction of the numerical error it still requires $n+1$ objective function calls to compute the partial derivatives to n design variables. This rapidly becomes impractical in 3-dimensional cases where cases have been explored with 100s of design variables [157, 158]. The adjoint methods for sensitivity calculations reduce the computation of the sensitivity to all design variables to a single flow solve.

This method, developed by Jameson [159] for inviscid flow and Nadarajah and Jameson for viscous flows [160], is based on optimal control theory; the flow equations are introduced as a constraint allowing variations of any number of design variables to be accounted for without recomputing the flow solution. The process can be derived in terms of Lagrange multipliers or duality in analogous manners [161]. This mathematical foundation highlights the relationship of the adjoint concept with sensitivity analysis.

The solution of the adjoint equations requires a bespoke solver which solves either the analytical adjoint or the discrete adjoint equations. The analytical adjoint is the adjoint of the analytical aerodynamic problem, it is the easiest way to implement the adjoint as well as the most computationally efficient. The discrete adjoint is calculated relative to the discretised flow equations, this method is more expensive, has a significant memory overhead [161] and is harder to implement. Despite these drawbacks some work suggests that the discrete method produces more accurate gradients for aerodynamic optimisation [162]. This is not altogether surprising as analytical adjoints provide the numerical adjoint to the exact flow equations, the discrete formulation takes into account influence of the mesh.

2.4.2 Aerofoil Parametrisation

Parameterisation methods ensure the interface between the optimiser and the CFD process in an optimisation framework. Their role is to translate the design variable vector into a geometry suitable for flow analysis. An effective parameterisation method must limit the optimiser to meaningful designs while having sufficient degrees of freedom to represent a varied design space. The use of gradient based optimisers necessitates the geometric design space defined by the parameterisation to be smooth, as it ensures the design variables used by the optimiser have smooth and continuous gradients. Ideally the smoothness of the parameterisation would be linked to proportionate changes in the flow but this is hard to guarantee due to the complexity of the Navier-Stokes equations.

2.4.2.1 Considerations for the design of a parameterisation method

Choosing a parameterisation method requires careful consideration of its impact on dimensionality, design space modality and geometric range. These properties have a direct effect on computational cost and the ability of an optimisation method to recover the global optimum. Previous systematic investigations by Vassberg et al. [8, 163] have highlighted the impact of dimensionality on the drag minimisation of a standard test case, showing the importance of geometric flexibility while maintaining a compact set of design variables. This need for compactness is independent of the cost of gradients as the number of design variables also impacts the number of optimisation steps. Indeed the cost of optimisation using quasi-Newton methods with adjoint methods is proportional to the number of design variables [8].

Work by Castonguay and Nadarajah [164], and more recently by Masters et al. [9, 57] provide comprehensive geometric and aerodynamic comparisons of the parameterisations presented earlier. The importance of a smooth geometric sensitivity was shown by Masters et al. [9]; the ability of a parametrisation to produce oscillatory shapes was detrimental to the overall performance in optimisation.

As aerodynamic behaviour is very sensitive to oscillations and the introduction of sharp corners, these behaviours must be limited by aerodynamic parameterisation methods. The properties needed by a parameterisation must translate design intent between the optimiser and the flow solver; if modes are not physically meaningful optimisation will be unsuccessful. In order for this to be possible, an effective parameterisation method must mirror the properties of the underlying objective and constraints.

These studies provide helpful guidance for the design of new parameterisation methods. The evidence shows that a parameterisation should be compact and provide smooth control. Oscillatory modes are very detrimental to progression through the aerodynamic domain [9], however this behaviour can be alleviated with intelligent scaling parameters. One of the key questions that appears in all methods is the desirable support radius of the basis function, i.e. the distance of influence. The support of certain parametrisation methods can be tuned (B-Splines, RBF domain elements) and it has been shown to have an impact. Global polynomial modes are usually undesirable but smoothness can only be maintained with a sufficiently large support radius. Kulfan and Bussoletti [165] suggest a number of other desirable properties for parameterisation methods including: intuitiveness; systematic process, the ability to represent many curves with the same process; and robustness to standard geometric transformations.

2.4.2.2 Constructive methods

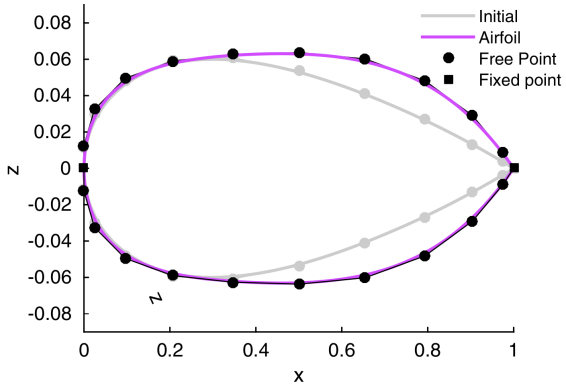
Constructive methods define completely the profile from the set of design variables; the parameters build the surface profile of the aerodynamic body. These include B-Spline (Figure 2.14a) and polynomial interpolation [166] in general, and Kulfan's Class Shape Transforms (CST) [165] and Sobieczky's PARSEC [167] in particular. The PARSEC method provides an intuitive parametrisation by building a polynomial from 9 physical aerofoil features, unfortunately this method has shown its limitations and is not adequate for modern optimisation problems [9, 57, 164].

The CST method represents shapes by a series of Bernstein polynomials (a special case of B-Splines) superimposed on a shape function. This method allows the polynomials to be well behaved at the leading and trailing edge of aerofoils [165]. Given appropriate shape functions this method can be adapted to work with a number of optimisation problems beyond aerofoils; the developers of the methods envisaged its use in engine nacelles, nose cones and wings. While capable of generating very complex shapes the CST requires expert configuration and has a limited ability to transition between classes of shapes and no ability to change topology. Beyond these two methods, a range of cubic B-Splines [168] and non-uniform rational B-Splines (NURBS) [169] have been used as direct methods to parameterise aerofoils.

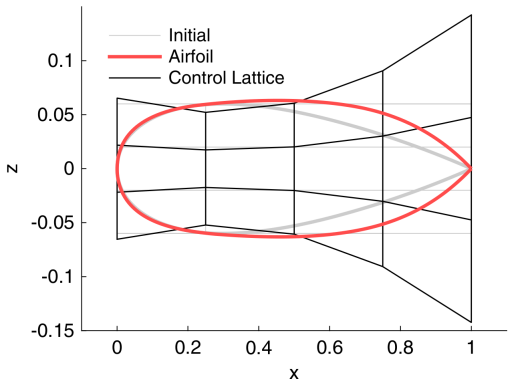
2.4.2.3 Deformative methods

Instead of generating an entire geometry, deformative methods define a set of modifications to a baseline geometry. These have the benefit of being easily adaptable to different cases. The simplest and most intuitive method is to use an existing discretisation as parametrisation, this means the position of each individual surface mesh vertex is used as a design variable. While this does not restrict the range of geometries that can be represented it tends to generate very multi-modal design spaces; it is desirable to have a smoother parameterisation. Historically the first smooth deformative parameterisation method is the Hicks-Henne bump functions [170]. This method specifies changes to the profile through a linear combination of sinusoidal basis functions added to the original geometric shape.

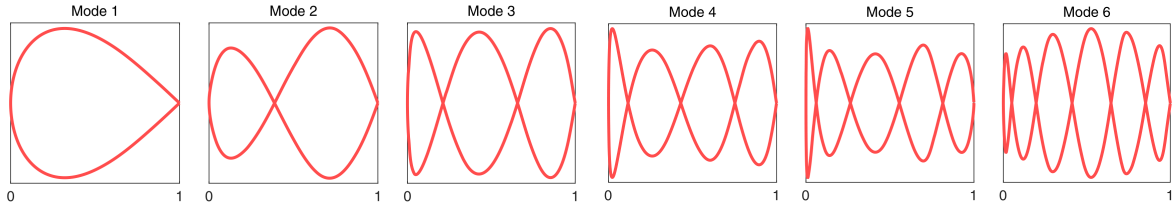
Free-form deformation (FFD) (Figure 2.14b) methods were developed for aerodynamic parameterisation [171, 172] and mesh deformation from tools originally used in computer generated graphics. These methods use a set of control points to define volume deformations to be applied to an existing profile. Two distinct methods exist, the first



(a) Cubic B-spline of a NACA 0012.



(b) FFD using a 4 by 6 Bezier control lattice.



(c) First 6 SVD modes for a library of symmetric aerofoils.

Figure 2.14: Examples of aerodynamic parameterisation methods from Masters et al. [9].

one is based on a Bezier surface which is used to define a deformation of the domain. The choice of the lattice of control points is the main method for varying the number of design variables which limits the possible design variable combinations. The second of these methods is the Radial Basis Function (RBF) domain element method. Unlike the previous method it requires an exact recovery of the deformation of the control points. There is no restriction to the number of control points or the distribution, points can be placed on or off the surface to deform. These methods are particularly interesting as their extension to 3-dimensions is trivial.

Another deformative method of interest here is the singular value decomposition (SVD) method [173, 174] (Figure 2.14c). This method relies on the proper orthogonal decomposition (POD) of a library of aerofoils. This generates a set of modes which can be used to represent all aerofoils. This method was found to be very efficient in the number of design variables required for geometric flexibility [9, 57]. While classified here as deformative, this method can also be used in a constructive framework.

2.4.2.4 Exotic aerodynamic parameterisation methods

The adjoint method by dramatically reducing the cost of gradient evaluation has allowed new parameterisation methods with many more design variables. While the number of design variables ensures optimal shapes can be represented; it can adversely affect the complexity of the design space and the path of an optimiser. Recent work by Anderson and Aftosmis [175] and separately by Masters et al. [176] have explored the possibility of enabling convergence by refining the design variable space as the optimisation progresses. This approach has proved extremely successful. When applied to the aerodynamics design optimisation discussion group (ADODG) NACA0012 transonic case (case 1), the B-Spline subdivision refinement approach of Masters et al. reduced drag to 4.2 counts [176], an 88% reduction compared to the previous best results.

Berguin et al. [177] have attempted to develop a method for aerodynamic dimensionality reduction. Using ideas similar to those used in the SVD decomposition of aerofoil libraries a set of *principal directions* are extracted from the aerodynamic response surface. These directions are then used as design variables within an optimisation process. This process reduces the number of simulations required during an optimisation but it requires a large overhead calculation to compute the important directions. This is envisaged as a method to speed-up design cases where multiple rounds of optimisation are likely to be necessary.

Investigation of the different available parameterisations show that none of the established methods naturally handle topology changes. Work by Hall et al. [155] has explored the use of volume of solid (VOS)[3] information to produce geometries based on level sets. This method is derived from an approach by Prilepov et al. [178] to extract fluid boundaries in multi-fluidic simulations. The idea of using volume information to provide flexible parametrisation seems particularly promising as it naturally supports progressive grid refinement and opens the possibility of defining new types of intuitive constraints. It is also intimately related to very successful density methods of STO.

2.4.3 Aerodynamics Design Optimisation Discussion Group

There have been sustained efforts in ASO for the last 20 years with increasingly complex problems in 2 and 3-dimensions being tackled. These efforts led to the creation of many optimisation frameworks customised to specific cases with no way to compare performance of these frameworks on standard benchmark cases. The aerodynamics design

[3]Note on terminology: 'volume' is used to mean 'area' in two dimensions.

optimisation discussion group (ADODG) was set up to provide a forum for researchers to compare their methods through five representative test cases. The 5 cases are: the NACA0012 inviscid drag minimisation under geometry constraint [179]; the RAE2822 viscous drag minimisation under lift, moment and area constraints [180]; the twist optimisation of a subsonic inviscid rectangular wing; the viscous optimisation of the Common Research Model (CRM) transonic wing [181]; and finally the Wing-Body-Tail optimisation of that same model [182]. These cases permit the objective comparison of different optimisation frameworks and have been used extensively by researchers since 2014.

Case 1 has been the most studied[4] of all the test cases as it appears to be the simplest, being 2-dimensional and inviscid. Part of this interest is due in part to the surprising complexity of the case. It is now well documented that the aerodynamic solution is hysteretic with small changes in Mach number [9, 176, 186, 188], and that the design space close to the trailing edge is very sensitive leading to local minima [9, 188]. Researchers from many of these studies combined results in Destarac et al. [194] to discuss and highlight some of the pathologies that exist with this single point, inviscid transonic optimised case. Case 2 has also been studied extensively[5]. This case presents none of the difficulties of the NACA 0012 inviscid optimisation, a very similar case was found to be uni-modal by Chernukhin and Zingg [58]. This is likely due to the dissipative effects of viscosity which while making it a more complex aerodynamic problem, makes for a smoother objective function.

Similar interest has been shown for the 3-dimensional cases. The twist optimisation of case 3 has been tackled by a number of participants[6] and is used as a validation before the exploration of the more complex 3-dimensional cases. The optimisations of case 4 and 5, especially the multi-point optimisations are much more computationally expensive than the 2-dimensional cases. This has meant fewer investigations[7] and less variety in the range of frameworks tested; effectively almost all previous investigations have used a combination of FFD B-spline volumes in conjunction with SQP. This is due to the small number of research groups which have tackled those 3-dimensional cases. However despite the similarity in approaches Meheut et al. [197] have shown the difficulty in achieving the same results with different implementations. These findings highlight the complexity of Aerodynamic Shape Optimisation: the integration of parametrisation, grid

[4] ADODG case 1 was tackled in [154, 169, 176, 183–192].
[5] ADODG case 2 was tackled in [154, 169, 185–187, 189–191, 193].
[6] ADODG case 3 was tackled in [54, 189–191, 195, 196]
[7] ADODG case 4 and 5 were tackled in [154, 157, 158, 188, 197–199]

generation, flow solver and optimiser are all likely to significantly modify the recovery of an optimum, and can cause the creation of spurious local minima.

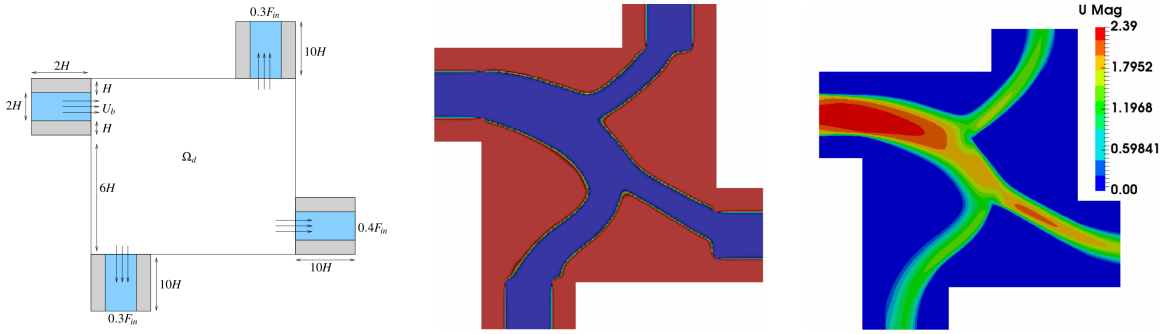
The benchmark problems have allowed significant progress through the study of different frameworks on the same problem [9]. This has allowed the publication of comparison studies with the participation of different research groups [188, 196, 197]. From this large number of studies some trends appear in the convergence of methods for each of the parts of the optimisation frameworks. In 2-dimensions the choice of parametrisation is fairly split as most of the more powerful methods have not reached maturity, however around 50% of the studies of Cases 1 and 2 use Kulfan's CST despite a below average performance in the latest comparison study [9, 57]. Overwhelmingly researchers use some form of SQP and often the SNOPT implementation. The use of gradient based optimisation is warranted as the modality of the design space is fairly well understood. The sensitivities are calculated using the adjoint method in almost all cases. A majority of researchers use a variety of structured codes, a notable exception have been studies using the cut-cell method by Aftosmis et al. [153].

In 3-dimensions the main parameterisation is the FFD method, which allows for good control of rotations and translations, this is unlikely to be optimal as it has a poor performance on 2-dimensional cases. SQP with adjoint sensitivity is also the optimisation method of choice for 3-dimensional cases however this is slightly problematic as some Eulerian cases have been shown to be multi-modal [58].

In an effort to answer the question of multi-modality in aerodynamic optimisation, case 6 was added to the suite of benchmarks used by the ADODG; it is the subsonic optimisation of a wing under a bending moment constraint. This case has been shown by a number of studies to exhibit multiple minima, especially as the design space grows [81, 200–202]. Poole et al. studied a similar case using niching approaches recovering multiple minima in a single optimiser pass [81].

2.4.4 Topological Optimisation of Fluid Flows

Progress in the field of topological optimisation of fluidic design is more limited and recent than in its structural counterpart. Both homogenization methods [204, 205] and level set methods [206] have been adapted to fluid topology optimisation in two and three dimensions. Modelling the Stokes equations and incompressible flows at low Reynolds numbers, solved by finite element elasticity solvers, these methods have yielded good results on the optimization of micro-fluidic devices and channel flows [205, 207]. In recent studies, Lattice-Boltzmann methods have been used to tackle some very low Reynolds



(a) Definition of a 2D flow manifold problem.

(b) Porosity values for the optimal solution.

(c) Velocity plots for the optimal solution.

Figure 2.15: Definition and topology optimisation of a 2D flow manifold for incompressible RANS flow with Reynolds number of 3500, images from Dilgen et al. [203].

number problems [208, 209]. Most of these methods rely on derivatives with regard to the porosity of the material to drive the evolution of the topology. Recent advances have seen broader ranges of turbulent incompressible flows being tackled [203, 210] (Figure 2.15). These methods do not maintain a smooth and crisp fluid boundary, limiting their use for compressible aerodynamics problems which use solutions to the Euler and compressible RANS equations at high Reynolds numbers. One notable exception is the cutFEM method by Villanueva et al. [211], where a LSM is used to parameterise shape and topology at very low Reynolds number.

Topological optimisation in fluid flow has seen active development in the field of micro-fluidic devices in Stokes flow [205, 207, 210, 212]. The assumptions of stokes flow of negligible inertial forces make it a very different problem to that of external aerodynamics. The governing equations are such that it can be solved by FEM and in such a framework solid bodies may be efficiently treated with infinite viscosity. This means that most of the findings in STO are readily transferred to the topology optimisation in Stokes flow but not to solutions of the Euler or Reynolds-Averaged Navier-Stokes (RANS) equations.

Hall et al. [141, 155] introduced topological flexibility for supersonic aerofoil optimisation and coastal defence barriers. Their method relied on a VOS parameterisation method and showed good initial results for both cases. It was particularly effective at generating multi-plane aerofoils to create cancelling shock patterns leading to very low wave drag. However the coupling with a gradient based optimiser [141] for the coastal defence barriers led to no changes in topology. While a good proof of concept the parameterisation by Hall et al. under-performs established parameterisation methods on shape

benchmarks; and its brittle formulation hinders the extension to 3 dimensions.

2.5 Summary

This chapter presented a comprehensive review of the fields needed to construct an effective topology optimisation framework for external aerodynamic flows. In optimisation, a clear trade-off was highlighted between the dimensionality, design space exploration, optimality and computational cost was highlighted. Topology optimisation was found to be an active field of research for incompressible flows, especially for low Reynolds number Stokes flow. For compressible aerodynamics applicable to external aerospace applications, optimisation frameworks use compact and smooth design spaces with local optimisation methods to enable fast convergence using expensive flow solvers with boundary fitted meshes as objective functions. Unfortunately, these properties come at the cost of the topological flexibility of the framework.

The only parameterisation which was shown to work for topology parameterisation for external aerodynamics is the volumetric level set parameterisation of Hall et al. [155]. A good proof of concept, its formulation is limiting: additional developments are required to enable the efficient topology optimisation. The next chapters of this thesis will explore the development of a new parameterisation methods for topology optimisation, suitable for aerodynamics, but flexible enough to tackle arbitrary problems. This new parameterisation will build upon the work of Hall et al. [155] and Masters et al. [9, 57, 176] to achieve topology parameterisation while maintaining efficiency close to that of existing ASO parameterisation.

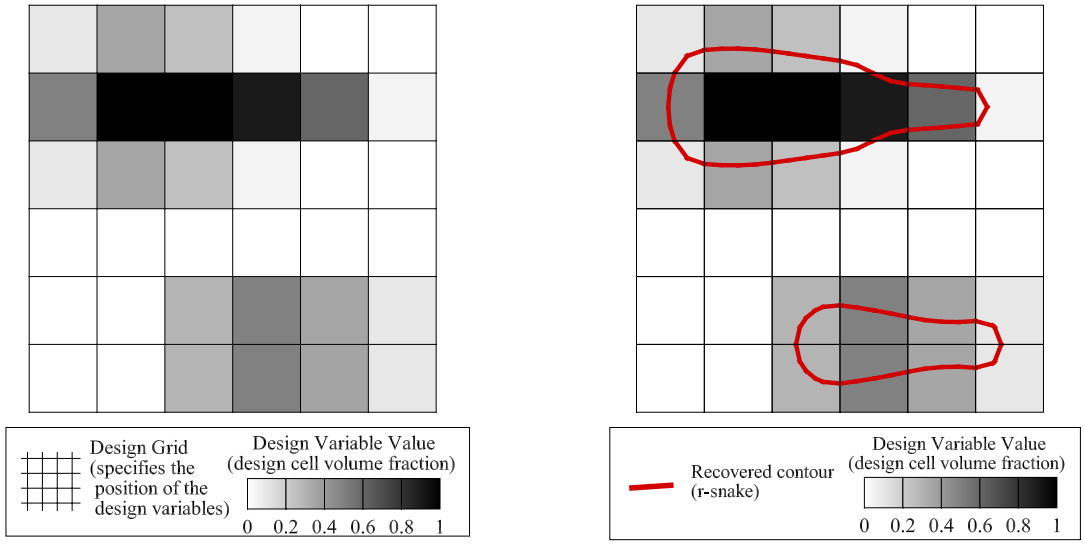
CHAPTER 3

DEVELOPMENT OF A PARAMETRISATION METHOD FOR TOPOLOGICAL OPTIMISATION

This chapter[1] presents development of the restricted snakes volume of solid (RSVS) method, an aerodynamic parameterisation that supports topological change while still performing efficiently on aerodynamic shape optimisation (ASO) problems (n.b.: the terminology 'volume' is used to also mean 'area' in two dimensions) . To be useful, the RSVS needs to fit into current modular aerodynamic frameworks, it must have: a sufficiently compact and smooth design space; be compatible with Eulerian finite volume CFD approaches; support adjoint gradients and be extensible to three dimensional problems. To ensure compatibility with the optimisation methods already shown to be effective in aerodynamic optimisation the set of design variables needs to be homogeneous; that is to say all design variables must be of the same type. This precludes the use of traditional aerofoil parameterisation methods with additional variables explicitly controlling the topology of the geometry. Development of this parameterisation requires the exploration contour representation and detection methods beyond the field of ASO, a brief review of the available methods is presented in Section 3.1.

The RSVS builds upon the volumetric aerodynamic parameterisation by Hall et al. [155, 214] which was an early topologically flexible parameterisation for external aerodynamics. Like the parameterisation of Hall et al., the RSVS uses volume of solid (VOS) design variables to control profile shape and topology, these were kept as they provide intuitive handling of topology change. However an effective contour generation

[1]This chapter is an extended version of parts of the peer-reviewed publication "Restricted snakes volume of solid (RSVS): A parameterisation method for topology optimisation of external aerodynamics" [213].



(a) Example design space for the RSVS with a set of design variable values specified

(b) Contour recovered from the design variables.

Figure 3.1: The RSVS provides a method for translating volume fractions specified on a grid (left) into a smooth profile which can have sharp features (right).

method has been developed to improve the geometric behaviour of the parameterisation. The main challenge in this type of parameterisation is the translation of the design variables into profiles suitable for CFD analysis. This chapter details how the RSVS provides a method to go from a VOS design space (Figure 3.1a) to a specific two body profile built for a set of VOS design variable values (Figure 3.1b).

In volume-based parameterisation the segmentation of volumetric information is done through a Cartesian grid, this means the design variables are best understood by a designer as grey-scale images on an underlying mesh (Figure 3.1a). This observation highlights the similarity between the parameterisation of geometries from volume information and the field of contour extraction in image analysis. Image segmentation, and medical image segmentation in particular, pose many of the same challenges as the volumetric parameterisation method considered earlier. The recovery of complex closed contours of arbitrary topology with limited computational expense is one that has been explored by the medical imaging community for the last 20 years. A class of methods for building such profiles that has seen significant and promising use is that of active contour methods [215, 216]. These methods rely on explicit vertex marching until the contour meets internal and external forcing conditions. Restricted snakes (r-snakes) developed by Kobbelt and Bischoff [217] are a type of parametric active contour designed

to handle topology changes efficiently. Section 3.2 shows how r-snakes are used in the RSVS to generate profiles of suitable aerodynamic quality that respect the values of VOS design variables. The shape of the r-snake is driven by a set of equations that were found to have desirable smoothness properties, these are presented in Section 3.2.1. This process was used to generate the profile in Figure 3.1b.

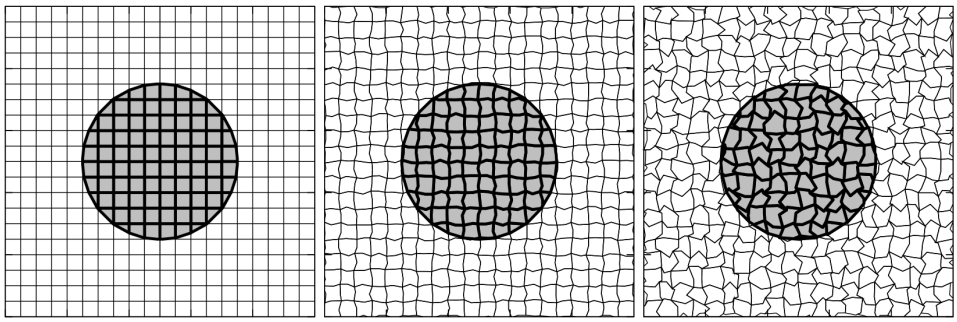
First, Section 3.1 reviews methods for the generation and control of boundaries and profiles from a range of fields of research. Section 3.2 presents the development and main features of a new topology parameterisation method suitable for aerodynamic applications: the restricted snakes volume of solid (RSVS). The implementation of this new method is validated in Section 3.3.1, where a range of geometries are generated, highlighting the compactness and flexibility of the method. Sections 3.4 and 3.5 provide details of the implementation of the RSVS parameterisation. Finally, Section 3.6, derives and validates the equivalence between profiles generated by the RSVS and non-uniform rational B-Splines of degree 2.

3.1 Contour Representation Methods

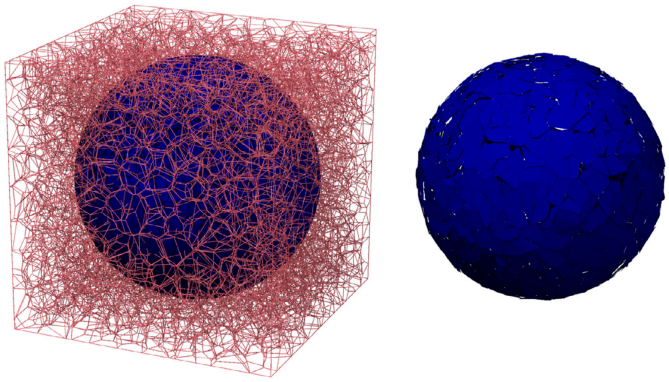
Current Aerodynamic parameterisation methods have been found to be insufficiently flexible for Topological Aerodynamic Optimisation. For this reason it is necessary to consider contour representation methods beyond those that have been used for ASO. Surface reconstruction is done in fluid simulations which involve multiple fluids with methods designed to extract interface boundaries and free surfaces from volume of fluid (VOF). Image segmentation, notably for medical applications, routinely deals with the need to extract topologically complex features in 2 and 3 dimensions. These fields use two main families of representation methods: volume representations and active contour models [216]. In this second class of approaches common methods include: parametric Active Contours, Level sets and B-spline curves. B-spline parameterisation is already in extensive use in ASO and will not be covered here.

3.1.1 Volume Representations

In multi-fluid simulations there is a need to define interfaces between fluids to accurately transfer pressures, and model surface tension forces. This boundary must be built from relative volume information between the different fluids specified on the discretisation of the fluid domain. The first method developed for this purpose was the simple line



(a) 2D PLIC of circular VOF specification from Lopez et al. [221].



(b) 3D PLIC reconstruction of a sphere from Lopez et al. [220].

Figure 3.2: PLIC reconstruction of circles and spheres from VOF information on irregular grids, images from [220, 221].

interface contour (SLIC) method which leads to a staircase profile aligned with the edges of the grid [218]. The quality of this surface was improved by the piecewise linear interface contour (PLIC) method which did not align the surface with the grid [219], however this still led to straight edges and discontinuous profiles as seen in Figure 3.2. Recent developments in VOF methods have extended PLIC-like methods to 3 dimensions and arbitrary grids but still do not guarantee continuous profiles [220, 221].

While these approaches have never been used “as is” for aerodynamic parameterisation, methods similar to VOF representation have been used for structural topology optimisation (STO). Density based methods such as solid isotropic material with penalisation (SIMP) discretise the profile in terms of densities of materials in each cell. As discussed in Section 2.2.1, these methods while very effective for topology optimisation in structures and Stokes flow do not maintain a surface representation. In industrial applications of density and homogenization methods, extraction of a manufacturable profile relies on stress-based redesign of structural skeletons [222, 223] or approximate interpretation of the density contours [224, 225]. These post-processing approaches to

profile generation do not have the required qualities of a good parameterisation identified in Section 2.4.2.1. While they are intuitive, they are not sufficiently compact and smooth for the design of external aerodynamic geometries.

3.1.2 Moving Fronts and Level Set Methods

The main alternative to snakes are level set representations of domain boundaries. In these, the boundary is implicitly represented as a level set of a function (the objective functional) defined over the entire analysis space. Osher and Sethian [226] developed the necessary mathematics and algorithms to predict and control the propagation of these fronts.

Their methods rely on propagation rules governed by a Partial Differential Equation (PDE) in which the internal and external constraints are encoded. The algorithm is based on Hamilton-Jacobi (HJ) formulations allowing the movement of domain boundaries with curvature dependent speed. This scheme was successfully applied to image segmentation in 2-dimensions by Chan and Vese [227], and in 3-dimensions plus time for fluorescent cell segmentation and counting by Dufour et al. [228].

These methods are very effective but can be less adaptable to new problems. These approaches are closely related to later efforts in structural topological optimisation using level set methods by the same author [229]. Level-set methods are not limited only to evolution using HJ formulations, but can use direct control of level set function parameters. Extensive use, especially in the Structural Topology Optimisation community, has led to a proliferation of formulations tuned to STO optimisation problems; these are discussed further in Section 2.2.2.

3.1.3 Parametric Active Contours

Parametric Active Contours are contour representation methods in which parameters explicitly define the position of a separation between domains. An example segmentation of a cell using a parametric contour defined by a spline is shown in Figure 3.3. This approach is different to implicit methods, where the boundary of a region is not directly accessible. Kass et al. developed an energy minimising spline, called a snake [230], which was attracted to image features. The movement of the snake is driven by the movement of control vertices called snaxels (portmanteau of snake-voxel), which are subject to internal and external forces. It was envisaged as a method for feature extraction, motion tracking and stereo matching.

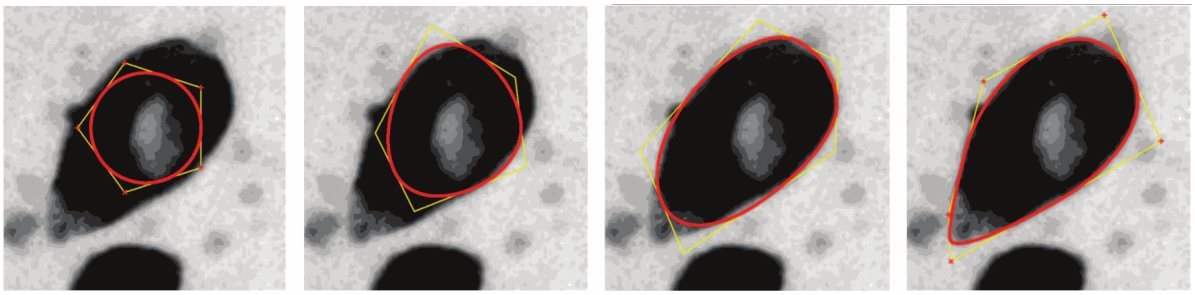


Figure 3.3: Evolution of a parametric active contour with 5 control points from initialisation at the centre of a biological cell, until it precisely segments the cell. Image from Delgado-Gonzalo et al. [216].

Snakes were later extended to recover objects of flexible topology by McInerney and Terzopoulos [215]. This was achieved by interpolating the snake on an Affine Cell Image Decomposition (ACID) , i.e. a tessellation using convex polytopes. The resulting snakes on ACID (or T-snakes) where able to merge and cut efficiently to match the topology of the boundary being recovered. This approach was applied to 2 and 3-dimensional medical images, successfully isolating complex features such as brain regions and vertebrae.

The snaking process was further simplified by Bischoff and Kobbelt [231] by restricting the snaxels to move along the grid edges. This approach of restricted snakes (r-snakes) made the detection of topology merging and separation trivial, removing the need to compute intersections between grid and snake. Collisions between different part of the profiles are reduced to a single comparison between scalar double values. While only developed in 2-dimensions by the author, extension to 3-dimensional convex grids can be done.

The strength of parametric sections is their ability to merge arbitrary external influences and self imposed properties to reach a desirable profile. Indeed snakes were designed to recover a wide range of image features while minimising internal forces [215]. This flexibility means they can be used in many applications and can respond to various external forcing, the original implementation supported real time user forcing [230]. The benefit of these approaches is that the external forcing can be treated as an external black-box process [231], allowing many possible conditions to be experimented with.

3.2 Geometry and Topology Generation using Restricted Snakes

The role of the parameterisation method is to provide an interface between an optimisation method and a solver to form a shape optimisation framework. Efficiency and flexibility of shape optimisation frameworks is limited by the geometric capability of the parameterisation method. This section presents how the restricted snakes volume of solid (RSVS) parameterisation translates sets of volume fraction design variables specified on a fixed grid into closed contours of varying topology. For optimisation frameworks to exploit the RSVS efficiently, this process must reliably produce smooth features at a resolution below the grid on which volume of solid (VOS) values are defined

To achieve the required level of smooth control, the RSVS profile is defined as the closed contour of minimum arc-length that will match the volumes of the design variables; it is built using a restricted snake (r-snake). The r-snake is a method for “vertex marching" which allows efficient topology handling and is tolerant of any layout of VOS design variables. The r-snake is a type of parametric active contour originally developed by Kobbelt and Bischoff [217]. This section develops the integration of the r-snake with the RSVS condition of minimising the arc-length under volume constraints. This condition was found to reliably produce smooth profiles enabling a compact parameterisation. Later sections explore the analytical properties of RSVS equations to show that the stated smoothness and compactness targets have been achieved. The RSVS process is summarised in Figure 3.4.

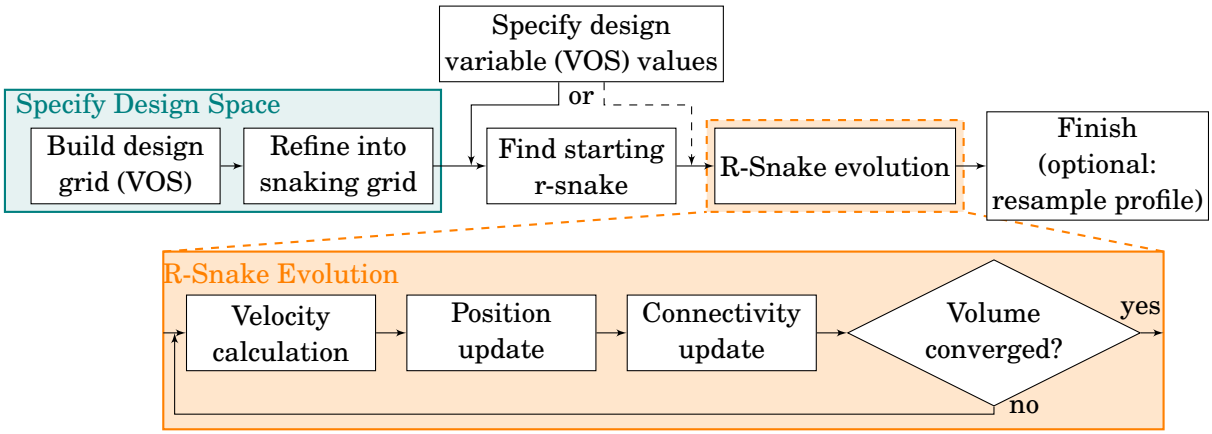
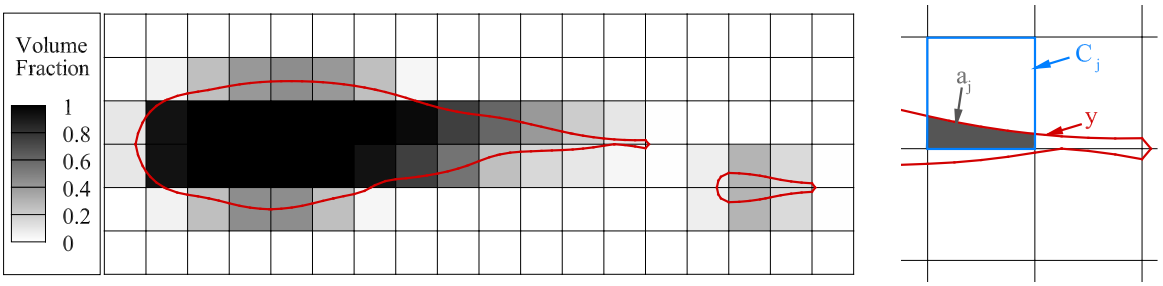


Figure 3.4: Flow chart summarising generation of a profile using the RSVS parameterisation method.



(a) VOS design variables as grey-scale and RSVS profile in red; 1 corresponds to a completely full cell and 0 an empty cell.

(b) VOS definitions for Equation 3.1

Figure 3.5: Example RSVS profile and design grid with label definitions for the governing equation (Equation 3.1).

3.2.1 Formulation of the RSVS

One of the difficulties in designing a parameterisation with topological flexibility is to maintain smooth control close to topology changes, as these are geometrically discontinuous regions of the design space. To define a set of VOS variables a grid is superimposed on the design space, where the design variables become the fraction of each cell within a geometry built from this information. This process is shown for a simple grid in Figure 3.5. This parameterisation procedure provides intuitive handling of topology change without maintaining explicit control of it. It is important that topology is not controlled explicitly as this would lead to a severely discontinuous design space which would not be usable with many of the traditional local and global optimisers used for aerodynamic optimisation.

The VOS design variables do not include in themselves rules for building a profile. These rules must generate profiles which are continuous and smooth, allow features smaller than the VOS design variables, and be indifferent to the type of grid they are being applied to. This last requirement opens up the possibility of using non-square grids for improved flexibility and compactness of the method. The rules must also be extensible to the generation of water-tight surfaces in three dimensions with minimal modification.

The condition used to define the RSVS is minimisation of the profile length, with the constraint that the area enclosed by the contour within each design cell must exactly match the value for of the VOS. The mathematical formulation of this problem is given in Equation 3.1. This system is analogous to the effect of a tensile force “shrink-wrapping" the required VOS in each cell; the benefit is it allows for smooth profiles in most cases but

can also recover sharp corners where the VOS requires it. In addition, this formulation is defined for any arbitrary layout of VOS variables; this means that anisotropic refinement of the mesh can be used to allow fine-grained control of the geometry without the need to increase the resolution throughout the design space. Length minimisation and area constraints naturally extend in three dimensions as the surface area minimisation under volume constraints of a geometry.

$$\begin{aligned} \min \quad & \oint \sqrt{1+y'^2}\,dx \\ \text{s.t.} \quad & \oint \left(y \cap C_j\right) dx = a_j \qquad \forall j \in \{0, \cdots, m\} \end{aligned} \tag{3.1}$$

In the expression above y is the closed profile built by the RSVS, m is the number of VOS cells in the design space, a_j the value of the VOS and C_j the outer boundary of the j^{th} cell. These are represented graphically in Figure 3.5b. The VOS is taken as a constraint on the area enclosed in both the profile and each cell. The next sections detail how this mathematical program can be solved using restricted snakes to produce an effective shape and topology parameterisation method.

3.2.2 Analytical Analysis of the Formulation

The formulation of the RSVS presented in Equation 3.1 can be analysed using calculus of variations to arrive at analytical expressions for its solutions. This analysis shows that the curves generated are continuous splines made of one arc of circle in each VOS cell. To make the derivation of these curves straightforward it is useful to consider the simplified case of a curve minimising length between two points of coordinates $(\alpha, 0)$ and $(\beta, 0)$ with a single volume constraint of value A. This continuous, single constraint problem is presented in Equation 3.2. The expansion to an arbitrary number of constraints is performed in Section 3.6.

$$\begin{aligned} \min \quad & \int_\alpha^\beta \sqrt{1+y'^2}\,dx \\ \text{s.t.} \quad & \int_\alpha^\beta y\,dx = A \end{aligned} \tag{3.2}$$

This minimisation problem needs to be solved for the expression of the curve $y(x)$. The optimum is the solution to the Euler-Lagrange equation of this problem which is expressed in Equation 3.3.

$$\frac{\partial}{\partial y}\left(\sqrt{1+y'^2}-\lambda y\right)+\frac{d}{dx}\frac{\partial}{\partial y'}\left(\lambda y-\sqrt{1+y'^2}\right)=0 \quad \Rightarrow \quad -\lambda-\frac{d}{dx}\left(\frac{y'}{\sqrt{1+y'^2}}\right)=0 \tag{3.3}$$

Equation 3.3 is integrated with respect to x in Equation 3.4 and rearranged to express y' as an explicit function of x in Equation 3.5.

$$-\lambda x-\left(\frac{y'}{\sqrt{1+y'^2}}\right)=-c_0 \quad \Rightarrow \quad \pm(\lambda x-c_0)^2=\frac{y'^2}{1+y'^2} \tag{3.4}$$

$$y'=\sqrt{\frac{\pm(\lambda x-c_0)^2}{1\mp(\lambda x-c_0)^2}} \quad \Rightarrow \quad y'=\frac{(\lambda x-c_0)}{\sqrt{1-(\lambda x-c_0)^2}} \tag{3.5}$$

For $y(x)$ to be a real valued function $y'(x)$ must be real: as a consequence only the positive case of Equation 3.5 is considered. Finally an expression for $y(x)$ is calculated by integrating with regard to x into Equation 3.6.

$$y=c_1\mp\frac{1}{\lambda}\sqrt{1-(\lambda x-c_0)^2} \tag{3.6}$$

Constants c_0 and c_1 are integration constants, λ is the Lagrange multiplier, these variables are to be chosen based on limit conditions. The equation for $y(x)$ is found to be the equation of a circle in Cartesian coordinates. To understand the effect of the various integration coefficients it is useful to rearrange Equation 3.6 into the canonical form of the circular equation.

$$(x-c_x)^2+\left(y-c_y\right)^2=\frac{1}{\lambda^2} \quad \text{with}: \quad c_x=\frac{c_0}{\lambda} \text{ and } c_y=c_1 \tag{3.7}$$

3.2.2.1 Multi-Constraint Cases

To be a useful parameterisation multiple constraints need to be specified in different regions of the design space. The expression for the length minimisation under many area constraints becomes Equation 3.8.

$$\begin{aligned} \min \quad & \sum_{i=0}^{m}\int_{\alpha_i}^{\alpha_{i+1}}\sqrt{1+y_i'^2}\,dx \\ \text{s.t.} \quad & \int_{\alpha_i}^{\alpha_{i+1}} y_i\,dx=A_i \quad \text{for } i\in\{0,\,\cdots,m\} \end{aligned} \tag{3.8}$$

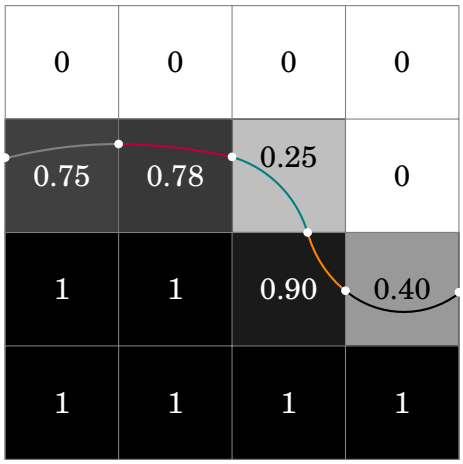
The steps followed for a single constraint are repeated for multiple constraints: the Euler-Lagrange equation is derived for this problem. The resulting equation reduces to a system of equations with one per y_i function:

$$\forall j \in \{0, \cdots, m\} : \quad \frac{\partial}{\partial y_j}\left(\sum_{i=0}^{m}\sqrt{1+y_i'^2} - \sum_{i=0}^{m}\lambda_i y i\right) + \frac{d}{dx}\frac{\partial}{\partial y_j'}\left(\sum_{i=0}^{m}\lambda_i y_i - \sum_{i=0}^{m}\sqrt{1+y_i'^2}\right) = 0 \tag{3.9}$$

$$\forall j \in \{0, \cdots, m\} : \quad -\lambda_j - \frac{d}{dx}\left(\frac{y_j'}{\sqrt{1+y_j'^2}}\right) = 0 \tag{3.10}$$

$$\forall j \in \{0, \cdots, m\} : \quad y_j = c_{1,j} \mp r_j\sqrt{1-\left(\lambda_j x - c_{0,j}\right)^2} \quad \text{with}: r_j = \frac{1}{\lambda_j} \tag{3.11}$$

Equation 3.11 means that the complete curve is a piecewise function made of arcs of circle. To compute an explicit solution to a given RSVS problem integration constants r_j, $c_{0,j}$ and $c_{1,j}$ need to be defined. The unknown constants can be solved for using three known properties of the final function: the VOS constraint, continuity and first derivative continuity. C_0 continuity is the result of needing the derivative y' of the curve to be defined at all points for the arc length to be defined. C_1 continuity is one of the assumptions of the Euler-Lagrange process (Equation 3.9) used to solve the system. Figure 3.6 shows an example analytical solution to the length minimisation defining a RSVS profile.



Each coloured line to the left is an arc of circle.

Each white dot is a point where those patches meet and C_1 continuity is applied.

Figure 3.6: VOS grid and circular patches are an exact solution to the RSVS governing equation

The boundary conditions are expressed in terms of the integration coefficients $c_{x,j}$, $c_{y,j}$ and r_j. Formulas for the condition corresponding to the constraints, continuity and derivative continuity are presented, respectively, in Equations 3.12, 3.13 and 3.14.

$$\text{Area contraint}: \quad \int_a^{\alpha_{j+1}} y_j(x)\,dx = A_j$$

$$\left[c_{y,j}x\right]_{\alpha_j}^{\alpha_{j+1}} \mp \frac{1}{2}\left[(x-c_{x,j})\sqrt{-c_x^2+2c_{x,j}x+r_j^2-x^2}+r_j^2\tan^{-1}\left(\frac{x-c_{x,j}}{\sqrt{r_j^2-(c_{x,j}-x)^2}}\right)\right]_{\alpha_j}^{\alpha_{j+1}} = A_j \tag{3.12}$$

$$\text{Continuity}: \quad y_j(\alpha_{j+1}) = y_{j+1}(\alpha_{j+1})$$

$$c_{y,j} \mp \sqrt{r_j^2-\left(\alpha_{j+1}-c_{x,j}\right)^2} = c_{y,j+1} \mp \sqrt{r_{j+1}^2-\left(\alpha_{j+1}-c_{x,j+1}\right)^2} \tag{3.13}$$

$$\text{Derivative continuity}: \quad y_j'(\alpha_{j+1}) = y_{j+1}'(\alpha_{j+1})$$

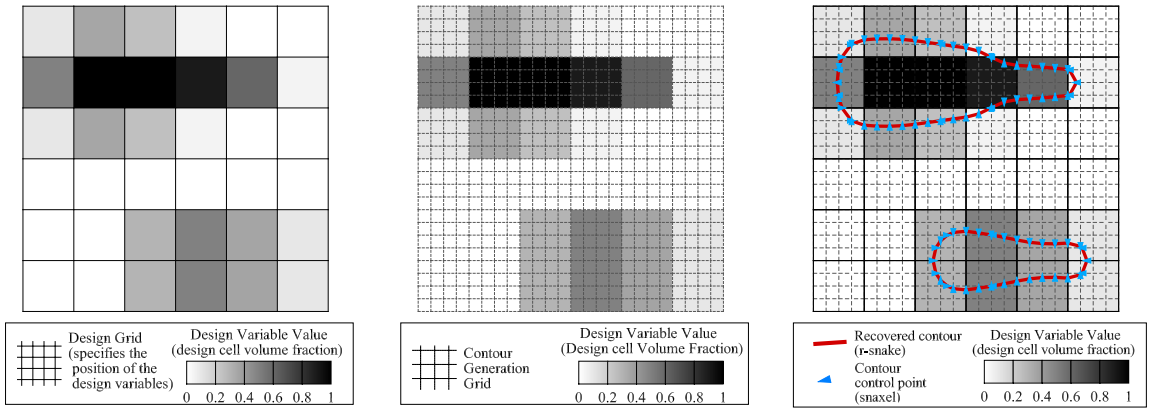
$$\frac{\left(\alpha_{j+1}-c_{x,j}\right)}{\sqrt{r_j^2-\left(\alpha_{j+1}-c_{x,j}\right)^2}} = \frac{\left(\alpha_{j+1}-c_{x,j+1}\right)}{\sqrt{r_{j+1}^2-\left(\alpha_{j+1}-c_{x,j+1}\right)^2}} \tag{3.14}$$

Unfortunately no explicit form for the three unknowns can be developed as a function of the values of the area constraints (A_j). Instead explicit analytical forms can only be found using numerical solutions to this system of equations.

3.2.2.2 Limitations of the Analytical Forms

To build the continuous profile, the points which separate each circular patch need to be found such that the profile and its derivative are continuous and the area constraints are fulfilled. Since the boundary conditions cannot be solved for analytically, an iterative process would be required. While the continuous solution exists, reliably finding it poses significant issues: positioning the knots, defining and changing the connectivity of the spline, detecting self intersections, and modifying the topology; would all require custom processes.

These difficulties with analytical solutions motivated the use of an existing profile marching procedure to find solutions to the RSVS governing equation. The profile is discretised and marched using a restricted snake: a type of parametric active contour specifically developed for efficient topological evolution. The following sections detail how this leads to a robust and reliable shape and topology control method from VOS data suitable for aerodynamic shape optimisation.



(a) Volume of Solid design grid (b) 4 by 4 Snaking sub-grid (c) Snake and snaxels recovering VOS information

Figure 3.7: The RSVS process for a simple two body profile on a 6 by 6 VOS grid (full lines); a 24 by 24 snaking grid (dashed lines); and the resulting length minimising discrete r-snake with snaxels on the edges of the snaking grid.

3.2.3 Topology Initialisation and Evolution Using a Restricted Snake

The previous section showed how length minimisation of a profile from coarse VOS information would generate smooth shapes. While analytical solutions exist they cannot be derived for arbitrary cases: a discretisation method is needed to generate and evolve the connectivity and topology of the profile. In the RSVS this role is fulfilled by a r-snake, a vertex marching procedure which is very efficient at handling self intersection and connectivity changes. This efficiency is achieved by constraining the control points of the contour to the edges of an underlying grid, allowing all intersections to be resolved as one dimensional cross-over of vertices. In this section only an outline of r-snake capability is presented; the snaking process is described more fully in Section 3.4; full implementation details are available in Bischoff and Kobbelt [217].

To build the RSVS parameterisation method the r-snake must be evolved until it solves the length minimisation problem specified in Equation 3.1. The resolution of the discretisation generated by the r-snake is controlled by the grid to which it is constrained; this implies that the smoothness of the generated profile is contingent on having a sufficiently fine grid. To allow the smoothness properties of the RSVS governing equation to be visible in the final profile, the VOS grid and *snaking grid* must be different: the snaking grid must be significantly denser than the design grid. This is to allow a high degree of geometric flexibility with few design variables: with this approach smooth

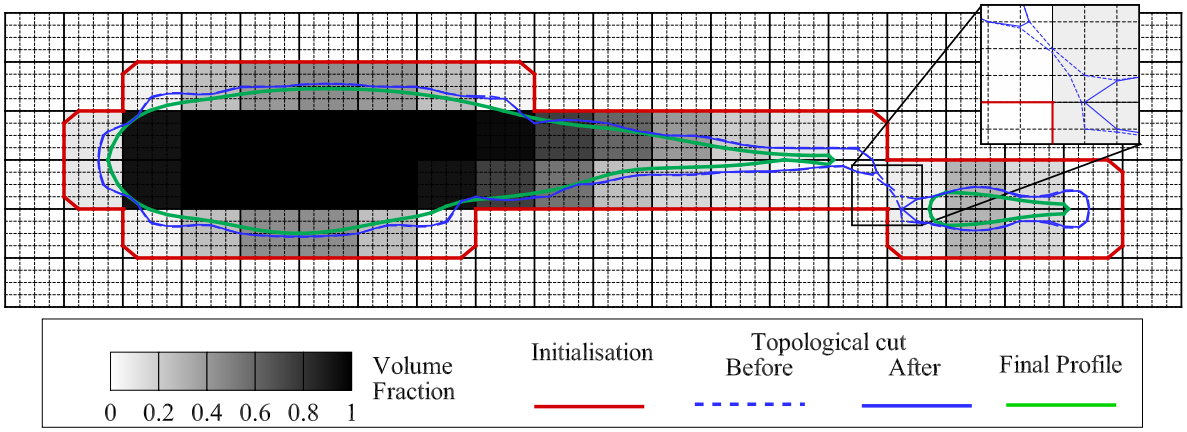


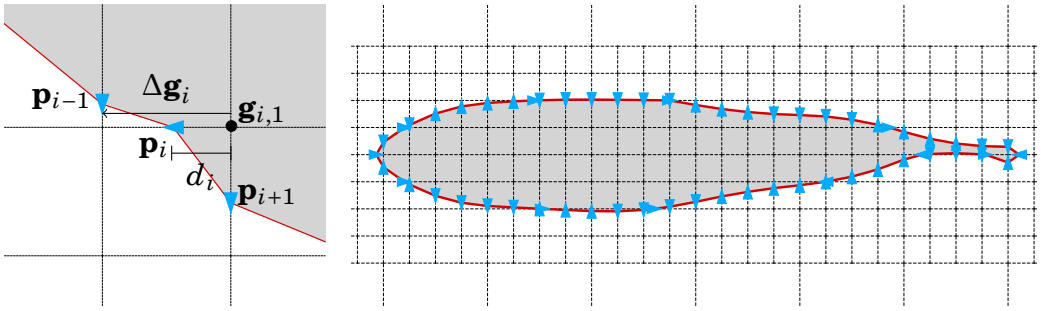
Figure 3.8: Evolution of the r-snake driven by the SQP algorithm solving the RSVS governing equation, showing 4 profiles: 1. initial (solid red, outer); 2. before topological change (dashed blue, intermediate); 3. after topological change (solid blue, intermediate); 4. final (solid green, inner).

features are recovered below the resolution of the VOS grid. This configuration of grids is presented in Figure 3.7.

While the RSVS rules define desirable properties for the final geometry, they do not specify a starting geometry. External aerodynamic optimisation is usually concerned with the design of outer surfaces, for this reason the r-snake is initialised at the outer boundary of non-empty VOS cells. An example of this type of initialisation is shown in Figure 3.8, by the outer red profile. This approach to initialisation has the benefit of always being defined and has an intuitive behaviour: it is similar to a force shrink wrapping the VOS design variables.

The benefit of using the restricted snake is that the topology can be modified if the volume fractions require it. This process is again illustrated in Figure 3.8 by the intermediate blue profiles. The dashed contour shows the r-snake before topology cutting and the solid blue line after the topology cut.

The topology change is handled efficiently by the restricted snake. The change in the geometry around the cut (shown in the close-up in Figure 3.8) is due to the r-snake algorithm removing invalid snaxel connections as specified by Bischoff and Kobbelt [231]. To maintain the integrity of the profile the r-snake algorithm limits the possible connections of a snaxel with its neighbours. The rules as developed by Kobbelt and Bischoff [217] are: no 2 connected snaxels can be on the same edge; snaxels must travel out of the profile. When two snaxels meet the profile connectivity is altered to by-pass them in what results in a change of profile topology. The connectivity rules are



(a) Close-up view of snake and snaxel

(b) Full view of a closed r-snake

Figure 3.9: R-snake contour (in red) with snaxels (in blue) evolving on the snaking grid (dashed line).

then applied removing the invalid connections that were generated.

3.2.4 Marching of the Restricted Snake

To drive the position of the r-snake the original continuous length minimisation problem (Equation 3.1) is discretised in terms of the r-snake and snaxel variables into the mathematical program in Equation 3.15. This discretisation process needs six properties from the r-snake geometry and the snaxel positions. The first three of these properties are part of the snaking algorithm; the last three properties of the snaxels are derived from connectivity and grid information, and are needed for the implementation of the discrete length minimization problem. These properties are: the snaxel index (i), used to reference it in all operations; the normalised position along an edge ($d_i \in [0,1]$); the scalar velocity along that edge ($v_i \in \mathbb{R}$); the snaxel position in Cartesian coordinates ($\mathbf{p}_i$); the direction of travel of the snaxel ($\Delta\mathbf{g}_i$) and the vertex of origin ($\mathbf{g_{i,1}}$); the normal vectors to the preceding and following edges ($\mathbf{n}_i$ and $\mathbf{n}_{i+1}$). These properties are represented graphically on Figure 3.9a.

$$\begin{aligned} &\min_{\mathbf{d}} \sum_{i=1}^{n} |\mathbf{p}_i - \mathbf{p}_{i-1}| \quad \text{with } \mathbf{p}_i = \Delta\mathbf{g_i} d_i + \mathbf{g}_{i,1} \\ &\text{s.t. } \mathbf{A(d)} - \mathbf{a} = \mathbf{0} \end{aligned} \tag{3.15}$$

In Equation 3.15 $\mathbf{d}$ is the column vector of all snaxel distances d_i, $\mathbf{a}$ is the column vector of target volumes a_j in each VOS cell and $\mathbf{A(d)}$ is the current volume in each VOS cell contained by the r-snake. The normalised snaxel positions ($\mathbf{d}$) are used as the design variables of the length minimisation problem. This formulation is very general,

it can be tackled on an arbitrary volume grid with any underlying convex snaking grid. This generality guarantees a high degree of flexibility in the range of shapes that can be represented. Beyond the wide range of design spaces that can be specified using the RSVS thanks to flexible grid requirements; the governing equation of the parameterisation can be solved for with any effective optimiser or solver.

To solve the discretised RSVS equation a method was required that would converge in few iterations and function evaluations; for this reason a gradient based method is a good choice. The availability of analytical first and second derivatives means that sequential quadratic programming (SQP) is a viable option. The form of the discrete length minimisation is relatively simple; the constraint is quadratic, and the objective behaves mostly quadratically with regard to vertex positions when the edges lengths are far from 0. This limited level of non-linearity allows a line-search to be avoided and the use of a damped Newton step. The damped Newton step is defined from a quadratic approximation to the discrete mathematical program; it is used to specify the snaxel velocities. The full derivation of the Newton step SQP is presented in Boggs and Tolle [49]. Only the final velocity update formula used in the RSVS is presented below (Equation 3.16).

$$\begin{aligned} \boldsymbol{\lambda}^{k+1} &= \left((\nabla_{\mathbf{d}}\mathbf{h})^T(\mathbf{H}_{\mathbf{d}}f)^{-1}(\nabla_{\mathbf{d}}f)\right)^{-1}\left(\mathbf{h}-(\nabla_{\mathbf{d}}\mathbf{h})^T(\mathbf{H}_{\mathbf{d}}f)^{-1}(\nabla_{\mathbf{d}}f)\right) \\ \Delta_{\mathbf{d}}^{k+1} &= \mathbf{d}^{k+1}-\mathbf{d}^{k} = -(\mathbf{H}_{\mathbf{d}}f)^{-1}\left((\nabla_{\mathbf{d}}f)+(\nabla_{\mathbf{d}}\mathbf{h})\boldsymbol{\lambda}^{k+1}\right) \end{aligned} \tag{3.16}$$

The change in distances $\Delta_{\mathbf{d}}^{k+1}$ is used as the velocities (v_i) of the snaxels, letting the snaking process handle damping and connectivity changes. The derivative terms required by this equation are: the Jacobian of the constraints ($\nabla_{\mathbf{d}}\mathbf{h}$); the gradient of the objective ($\nabla_{\mathbf{d}}f$) and the Hessian of the objective ($\mathbf{H}_{\mathbf{d}}f$). Thanks to the formulation of the snaking process all these values are available analytically by differentiating the appropriate area and snaxel position with respect to the design variable to the length minimisation program, the distances d_i.

One challenge imposed by the use of a r-snake is that the connectivity of the profile changes every time a vertex of the *snaking grid* is crossed. This causes a change in the number and layout of snaxels and their normalised distances d_i; changing the size and layout of decision variables of the length minimisation. The benefit of the Newton-SQP approach specified in Equation 3.16 is that there is no dependence on previous quantities. This means that there is no incompatibility brought about by changes in the design variable vector, and the constraint vector does not change.

Issues could arise if the Hessian was approximated using Broyden-Fletcher-Goldfarb-Shanno (BFGS); however calculation of the derivatives necessary to the evaluation of Equation 3.16 is done analytically. The derivation of the area and length derivatives are presented in Sections 3.5.3 and 3.5.3, respectively. These derivatives are necessary for the implementation of the RSVS.

3.2.5 Towards a Modular Approach to Parameterisation

The RSVS has been developed and implemented as a length minimisation problem solved using a parametric active contour and a Newton step SQP. These choices lead to a parameterisation effective for the representation of aerodynamic geometries. While the primary purpose of this thesis is to study the current implementation of the RSVS and develop aerodynamic shape and topology optimisation frameworks; it is interesting to consider how the geometric properties could be tailored with minor modifications to the parameterisation.

Indeed the RSVS uses a framework for volumetric parameterisation which can be partially modified without affecting other components. The components of this framework are (in parenthesis the current implementation):

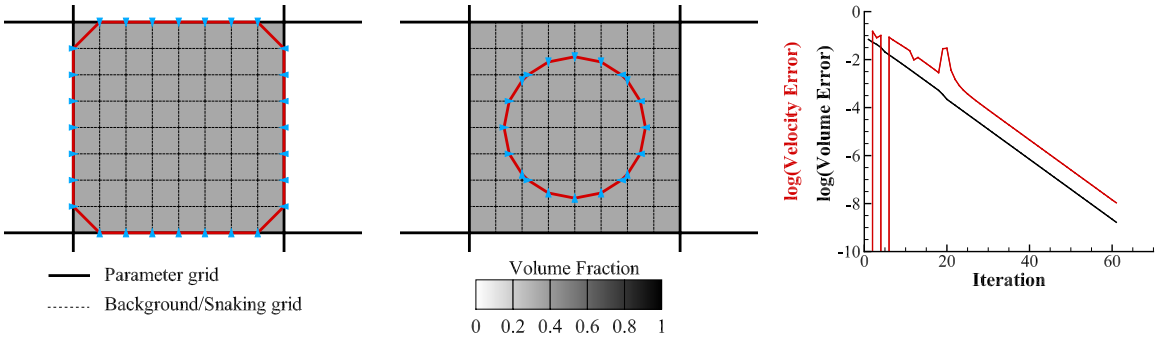
- the objective function (length minimisation);
- the profile discretisation (r-snake);
- vertex update procedure (Newton step SQP).

Each of these components can be swapped out if it did not fit with a desired application. For example increased smoothness requirements could be implemented by using a higher order objective, or the SQP could be replaced with a first order method to evolve geometries with millions of vertices in 3 dimensions.

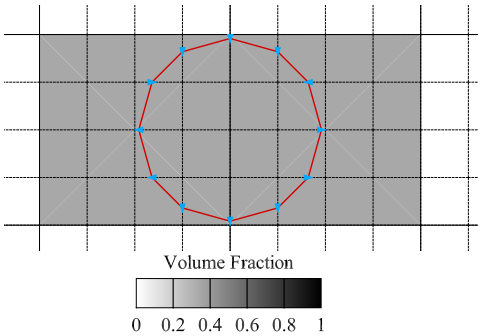
This flexible formulation means that the RSVS approach is more likely to be useful across a range of design optimisation problems. This is particularly interesting as the parameterisation requires a significant development effort; which can be amortised by using it for different applications.

3.3 Parameterisation Results

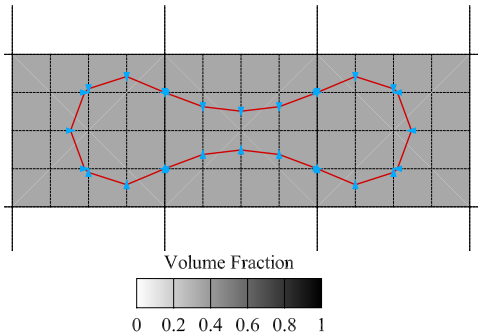
To ascertain the behaviour and performance of the RSVS methods, this section presents profiles generated using the RSVS parameterisation method. The profiles shown in this section were all designed with fixed volume fraction values to test the capability



(a) Contour defined by a single VOS cell, including initial (left), final profile (centre), and the convergence history (right)



(b) Contour defined by two volume fraction cells for $\mathbf{a} = 0.35$



(c) Contour defined by three volume fraction cells for $\mathbf{a} = 0.35$

Figure 3.10: Smoothness validation cases for one, two and three design variables.

of the parameterisation method; profiles generated by shape optimisation frameworks where VOS values are used as the design variables of shape optimisation processes are shown in Sections 5.3.2 and 5.3. Sub-Section 3.3.1 presents validation of the RSVS parameterisation while sub-Section 3.3.3 shows results for the geometric recovery of common aerofoil sections.

3.3.1 Validation Tests

This section presents some of the validation tests carried out on the parameterisation method to show that it does enforce the volumetric constraints and minimises the arc-length. The focus is on the convergence of the RSVS parameterisation in terms of the optimality of the r-snake as a solution to the discrete length minimisation. Cases with 1, 2 and 3 design variables (VOS cells) are shown in Figure 3.10; as expected these cases produced circular profiles. These results confirm that the SQP algorithm is able to cope with the design space generated by the underlying r-snake process and minimizes

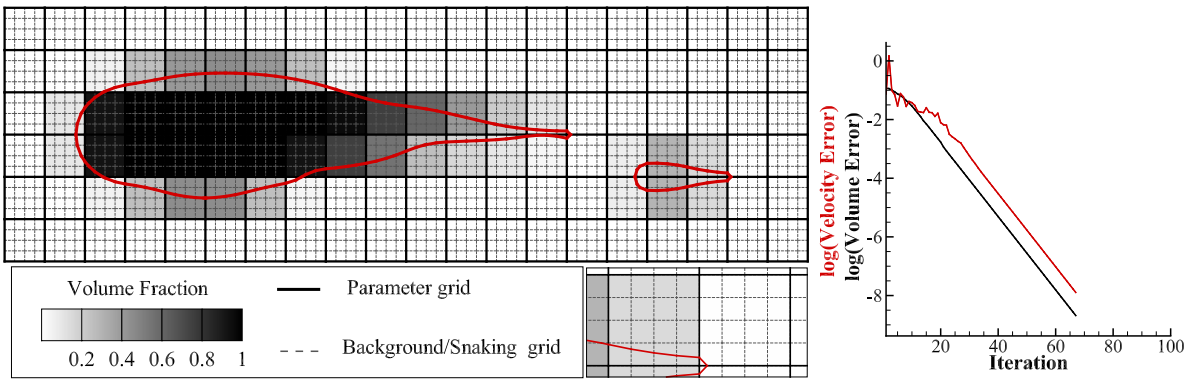


Figure 3.11: Multi-body aerofoil with parameter (volume) grid and background (snaking) grid as well as snake convergence history of the r-snake volume error and snaxel velocity.

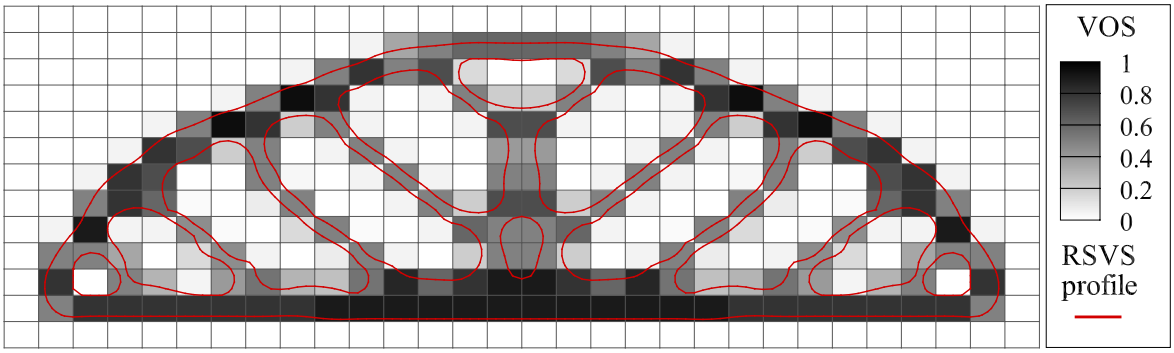


Figure 3.12: Bridge like structure manually generated with a 28 by 11 VOS grid.

the profile length for a given volume. This also validated the capability of the RSVS parameterisation to represent smooth features below the resolution of the VOS cells.

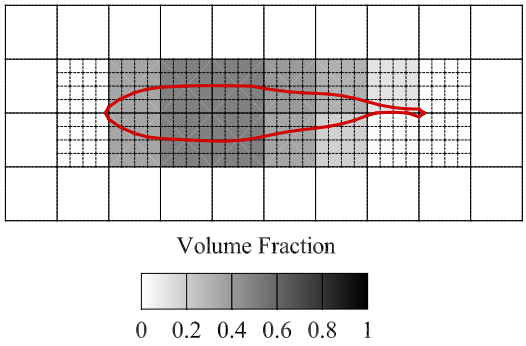
The geometric and topological flexibility of this parameterisation is demonstrated through Figure 3.11 for an aerofoil with flap. It was found that the number of snaking steps scales slowly with the number of volume cells (see Table 3.1). This is explained by the properties of the Hessian of the profile length and the Jacobian of the area constraints. Both have few off diagonal terms which means that interaction between snaxels and with the constraints is limited to those in close proximity. These properties mean that the algorithm is scalable and can be used to represent complex geometries with large numbers of snaxels and constraints. This is important as it ensures that the algorithm is capable of converging in few iterations for larger sets of design variables that could be necessary for complex topological optimisation cases. Beyond traditional aerodynamic shapes, the parameterisation is capable of handling internal voids and fine branching structures with few design variables, making it a potentially viable parameterisation in topology optimisation of other disciplines. This capability is shown in Figure 3.12, where

a suspended bridge is generated.

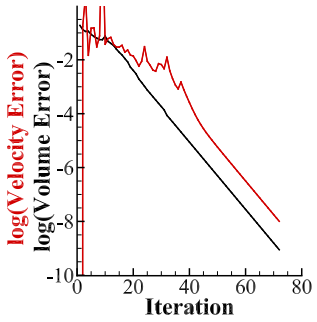
Figures 3.13, 3.14 and 3.15 illustrate the range of shapes that are possible with a small set of design variables. The 2 by 8 layout used to define these profiles is sufficient to control the upper and lower surfaces independently as well as the sharpness of leading and trailing edges. Importantly, even for the oscillatory profile smoothness of the profile is maintained below the resolution of the design variable. Figure 3.16 shows the topological flexibility does not impact the smoothness of the profiles.

One of the key benefits of this parameterisation method over previous VOS methods is its natural ability to build sharp corners and straight lines. As the required volume fraction is decreased at the edge of the profile, the minimisation of the profile length tends to create a very small feature which tends to a sharp corner as the volume fraction tends to 0. This effect can be seen in Figure 3.17, where both the leading and trailing edge are fixed in place by VOS values of 10^{-5} at the extremities. This use of small volume fractions to modify the properties of the curves is analogous to the introduction of a knot inside a spline. A smoother leading edge can be achieved by removing these volume fractions or by designing grids with more control at the leading edge, this is exploited for the geometric matching of aerofoils in Section 3.3.3. This ability to transition between sharp and smooth shapes is very important to the design of useful aerodynamic bodies which often require sharp trailing edges, leading edges or corners to control the point of separation. These small volume fractions can also be used to fix the length of a profile by effectively pinning leading and trailing edge position.

Most aerodynamic geometries are designed and optimised in a non-dimensionalised space; this requires parameterised profiles to be of unit chord. To control the leading and trailing edges position, a small VOS is maintained in the corner cells of the design space; these volume fractions pin the profile at those locations. This process is used in aerofoil optimisation cases to control of the chord length. While the position of leading and trailing edges can be enforced at VOS grid intersections, it cannot reliably be forced at other locations. This limitation is compounded by the fact the SQP algorithm converges fastest for isotropic Cartesian snaxel grids. This means that the range of thickness to chord ratios that can be explored efficiently by a small set of design variables is restricted. These limitations are overcome by introducing an external *grid aspect ratio*. This external parameter controls the stretching and shrinking of the profile generated by the snaking process. This stretching is applied after the snaking process to allow the snake to be generated on the most efficient grid and it is a way of varying the weight of each dimension in the calculation of the arc-length of the profile.

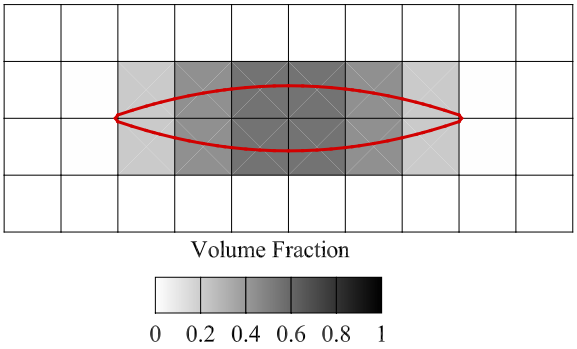


(a) Final Profile

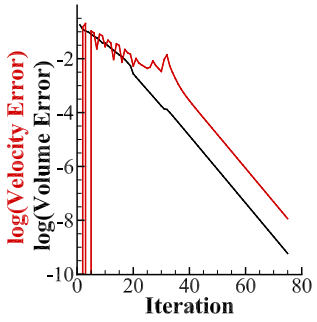


(b) Convergence history of profile

Figure 3.13: Pseudo-aerofoil reconstructed with 16 design variables

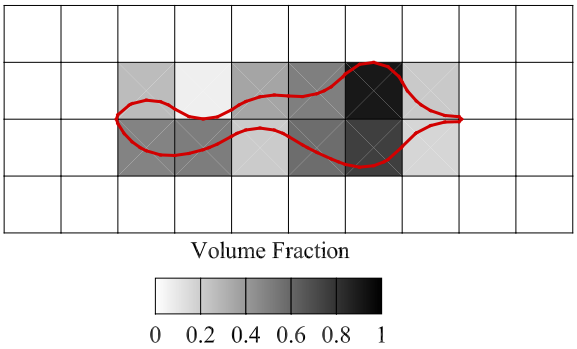


(a) Final Profile

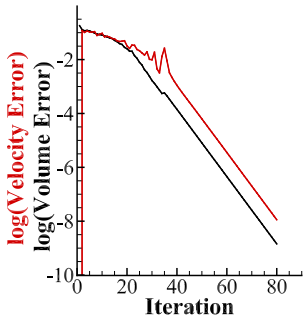


(b) Convergence history of profile

Figure 3.14: Paraboloid profile for low supersonic wave drag

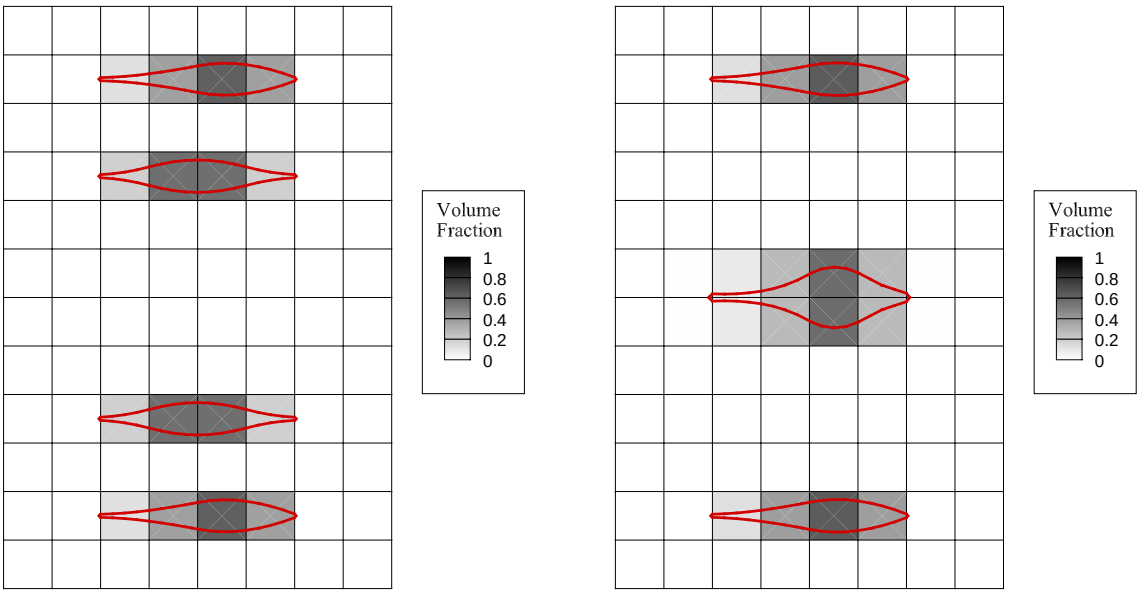


(a) Final Profile



(b) Convergence history of profile

Figure 3.15: Profile reconstructed by random design variables



(a) Quad-plane

(b) Triplane

Figure 3.16: Multi-plane profiles generated on a 10 by 6 design variable layout, these were generated automatically during supersonic flow optimisations discussed in Section 6.3.

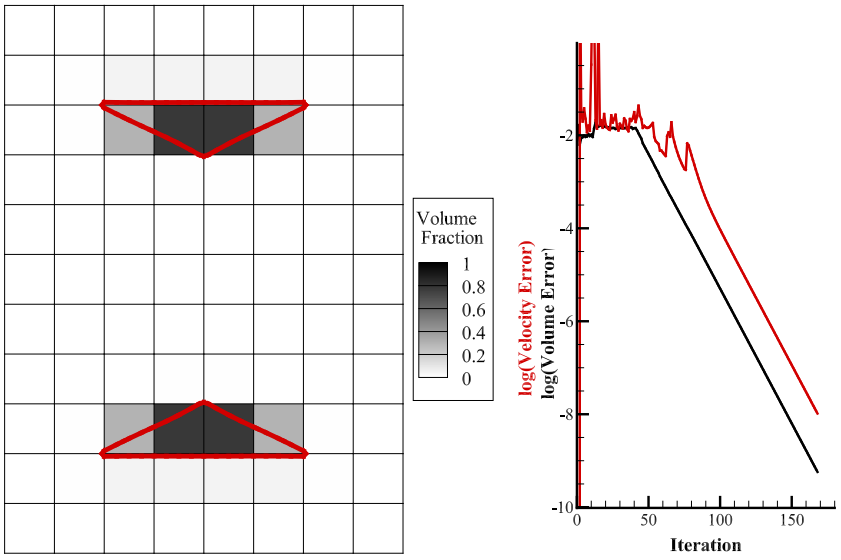


Figure 3.17: Reconstruction of a Busemann biplane [232] using 6 longitudinal design variables.

Table 3.1: Table presenting convergence data for Cartesian RSVS layouts.

Design variable layout	1x1	1x2	1x3	2x8	2x8	2x8	9x6	4x18
Figure	3.10a	3.10b	3.10c	3.13	3.15	3.14	3.17	3.11
Iterations to volume convergence ($e_{vf} < 10^{-9}$)	61	64	61	73	80	76	160	68

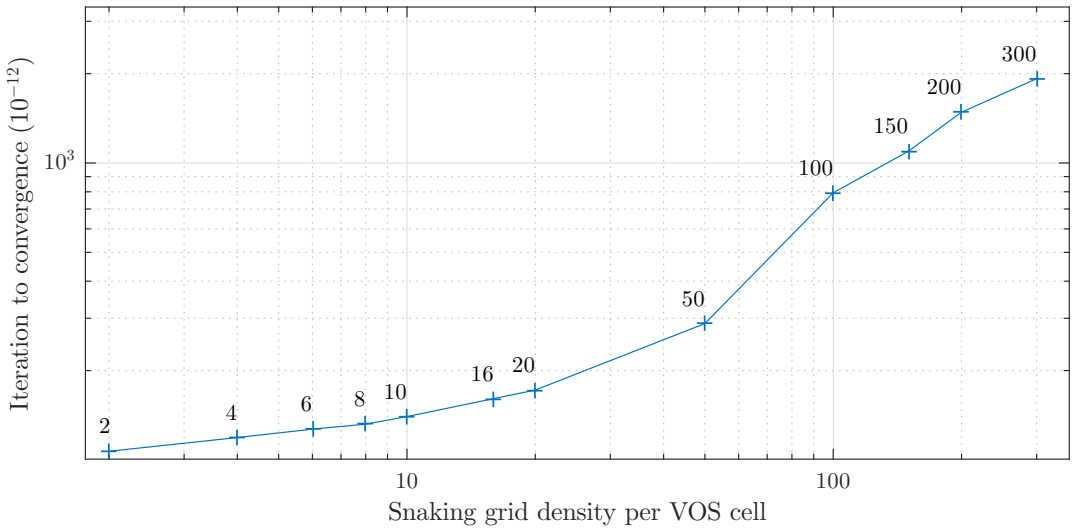


Figure 3.18: Convergence of RSVS profiles depending on the density of the snaking grid relative to the VOS grid.

3.3.2 Practical grid design for the RSVS

The key consideration in allowing good performance of the algorithm is the relation between the design grid (carrying the VOS information) and the snaking grid (over which the r-snake evolves). If the underlying grid is too coarse the optimisation process cannot converge as the combination of volume constraints and smoothness conditions makes for a very stiff system. If the underlying grid is too fine there is a significant computational cost increase. For most applications a cell refinement level of 4 (each volume cell is split into 16) yields good results. The full impact of snaking grid density is shown in Figure 3.18 where the impact of grid density on number of iterations is shown. Where high curvature is required within a single volume cell it can be desirable to increase the refinement level, however in most cases it is preferable to increase the number of volume cells as these afford increased geometric control.

The use of only Cartesian distributions of design variables will not be sufficient to achieve the fine shape control afforded by other parameterisation methods. Beyond

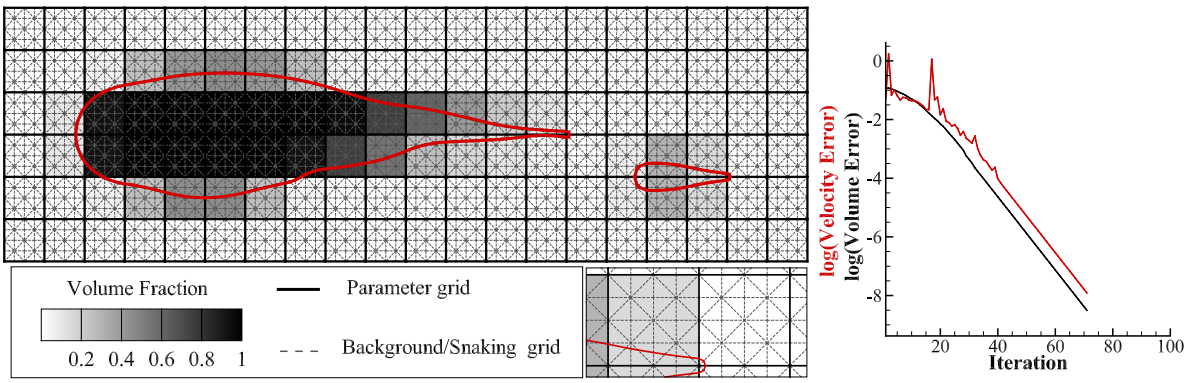


Figure 3.19: Profile generation on a triangular snaking grid (same VOS values as Figure 3.11).

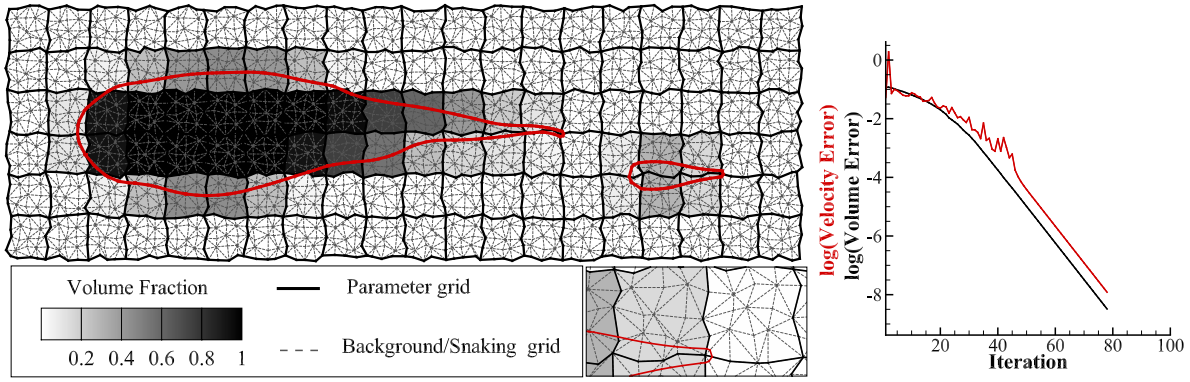


Figure 3.20: Profile generation on a VOS and snaking grid with noise (same grid topology as Figure 3.19).

those types of grids, the RSVS can be defined and evolved on arbitrary convex grids, with only a small convergence penalty. Figures 3.19 to 3.21 show profiles similar to the aerofoil with flap presented earlier built on non-Cartesian grids. These results highlight the wide range of design spaces that can be exploited by the RSVS, showing that this parameterisation can be tailored to any application with an appropriate VOS grid. Figure 3.19 shows the same aerofoil with flap evolved on a triangular snaking grid. To show that the RSVS recovers smooth surfaces even from non-smooth design spaces, noise was added to the snaking and design grids in Figure 3.20. Similarly a very similar profile is recovered a on a grid with triangular VOS grid (Figure 3.21), showing that the parameterisation does not suffer from an extreme grid dependence. The VOS grid can be deformed to take any arbitrary stretching to allow the design of specific property into the profiles generated by the RSVS. Figure 3.22 shows a grid with a sine wave added to the vertical axis. In the following section geometric recovery of aerodynamic bodies is performed; it highlights the benefit of manual grid tailoring afforded by the RSVS when

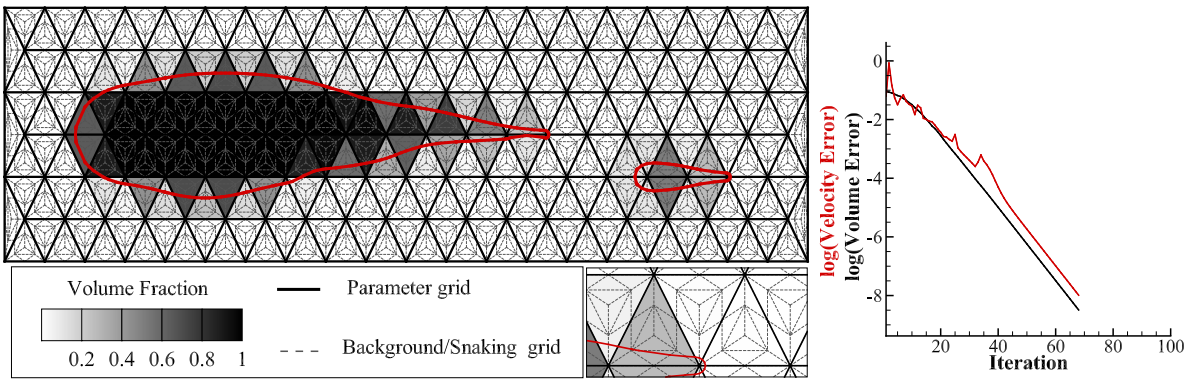


Figure 3.21: Profile generation on a triangular VOS grid (the profile is the same Figure 3.11).

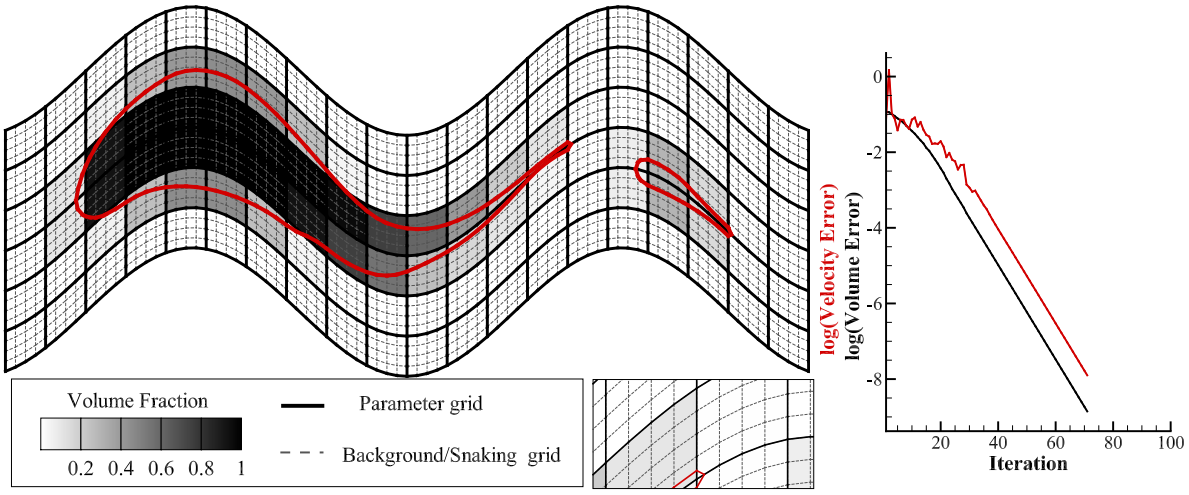


Figure 3.22: Profile generation on a VOS grid with a sinusoid component (same VOS values as Figure 3.11).

tackling a problem with specific, known, properties.

3.3.3 Geometric Inverse Design of Aerofoils

To validate the geometric flexibility of the r-snake parameterisation and its suitability for aerodynamic profile generation inverse design of aerofoils was performed. The profiles were evaluated against Kulfan's Wind Tunnel Tolerance (WTT) [165] using the same process as the one used in the review of aerodynamic parameterisation methods by Masters et al. [57]. The volume fraction values are specified on the RSVS grid by superimposing the profiles onto the grid and working out the intersection with each cell. Kulfan's WTT prescribes bounds in the maximum distance between profiles, given in

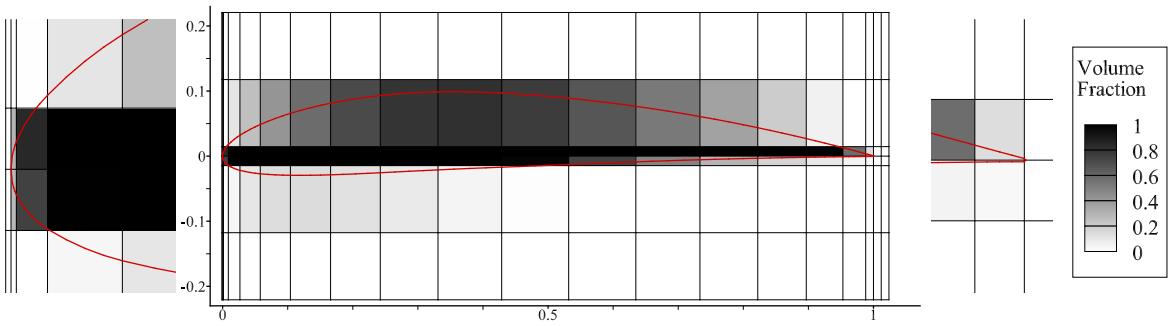


Figure 3.23: Inverse design of the NACA 4412 on a 17 by 4 anisotropic VOS grid.

Table 3.2: Result of the Inverse design of 4 aerofoils using 68 design variables

	NACA 0012	NACA 4412	ONERA M6	RAE2822
$\max\left(w_i y_{i,ERR}\right)$	2.301E-04	2.297E-04	3.590E-04	6.285E-04
Weighted RMS Error	4.123E-05	4.281E-05	6.195E-05	9.353E-05

Equation 3.17.

$$\text{WTT}: \quad \max\left(w_i\left(y_{target}-y_{approx}\right)\right) \le 8\times 10^{-4} \quad \text{with}: \begin{cases} w_i = 2 & \text{for } x < 0.2 \\ w_i = 1 & \text{for } x \ge 0.2 \end{cases} \tag{3.17}$$

In order to achieve this tolerance for aerofoils with a reasonable number of design variables a highly anisotropic VOS grid was used. A longitudinal distribution with cells clustered at the leading edge and at the trailing edge was devised to enable accurate positioning and definition of the leading and trailing edges. In the transverse direction cells were clustered close to the chord line to allow the lower surface to cross over without causing large interferences with the upper surface. The VOS grid with the design variable values and the r-snake is shown in Figure 3.23 for the case of a NACA4412 aerofoil meeting the Kulfan's Wind Tunnel Tolerance (Kulfan's WTT).

A detailed study of the inverse design using this VOS grid was performed on 4 aerofoils representative of common aerodynamic sections; these are the NACA 0012, the NACA 4412, the ONERA D aerofoil and the RAE 2822. The error values for these aerofoils are presented in Table 3.2. Kulfan's WTT is matched for each aerofoil using the grid in Figure 3.23. This grid was tested on a further 65 NACA aerofoils of which 63 were recovered to the Kulfan's WTT (97%), the averaged results for this second set are in Table 3.3.

While 68 design variables is more than the 20 to 30 design variables required by the established aerodynamic parametrisations studied by Masters et al. [9, 57] for this level

Table 3.3: Result of the Inverse design of 65 NACA aerofoils using 68 design variables

Population Values	Mean	$10^{\text{MEAN}(log(w_i y_i))}$	Median	Maximum	WTT satisfied
Weighted RMSE	5.573E-05	4.42E-05	4.137E-05	5.864E-04	
$\max\left(w_i y_{i,ERR}\right)$	6.179E-04	3.10E-04	2.613E-04	1.109E-02	96.9%

of fidelity; the number of active VOS design variables that need to be controlled in an optimisation is smaller than the total number of design variables in the grid used in this case. During the optimisation process only the design variables which contain the edge of the profile are of interest, this reduces the number of active design variables from 68 to 38 in the case of the NACA 4412 presented in Figure 3.23. This design variable reduction is then coupled with an overflow method which ensures smooth transition between design variables ensuring the optimisation framework only sees smooth geometric changes.

The largest errors appeared for the RAE 2822 aerofoil at the trailing edge, this is due to the very thin and curved trailing edge; this causes both the upper and lower surface to be contained in the same VOS cell which does not allow sufficient control. This case highlights the difficulty in building knowledge about a specific aerodynamic case into a very general and flexible parametrisation method. Rather than tuning the design grid to each case individually a generalised method based on local refinement of design variables is developed in Section 4.4. This hierarchical approach to parameterisation offers the possibility of the RSVS tuning itself to the requirements of a given optimisation problem.

3.4 Implementation of the Restricted Snake for the RSVS

The formulation of a new parameterisation method using VOS to build aerodynamic shapes has been presented in the previous sections. It relies on a topologically flexible parametric active contour to evolve and define a smooth geometry suitable for aerodynamic analysis. This section presents a detailed look at the implementation of the restricted snake, the process is summarised in Figure 3.24. To keep the relation between design variable and the profile intuitive, the contour recovery process and its formulation is explicit. This explicit relationship is achieved using a parametric snake to extract the profile from the VOS information specified by the design variables.

A parametric snake is composed of connected vertices called snaxels travelling freely

over a grid. The movement of these snaxels is governed by a number of simple rules that allow a large range of shapes to be represented. The development of a robust and flexible process for the movement of snaxels is critical to the efficiency and usability of the entire parameterisation method. The following sections contain the general ideas of the snaking process as well as details that separate this work from previous implementations. The current work uses the algorithm developed by Bischoff and Kobbelt in [217] for restricted snakes (r-snakes). Parametric snake contour recovery is presented in this section with the specific purpose of developing a VOS parameterisation method in mind; for a general and complete description the reader is referred to the work by Kobbelt and Bischoff [217].

3.4.1 Snaxel Mechanics

Snaxels are the control points of the active contour, their evolution follows three main steps: movement along edges; splitting (or breeding) when a grid vertex is encountered; and deletion when an illegal connection is formed. These three basic steps are sufficient to let snakes evolve on the grid, additional processes are necessary to avoid self intersecting profiles. These are explained in more detail in the next subsections.

3.4.1.1 Definition and properties

An understanding of the properties defining snaxels is necessary to the implementation of the RSVS parameterisation. The snaxel definition below comprises connectivity information with other snaxels, and inheritance information relating to the snaking grid to which it is constrained. Both types of information are necessary to the computation of the derivatives used in Section 3.2.4 to drive the position of the snaxel.

- The snaxel index (i), used to reference it in all operations;
- The normalised position along an edge: $d_i \in [0,1]$;
- The scalar velocity along that edge: $v_i \in \mathbb{R}$;
- The edge index within the grid (can be replaced by the arrival vertex);
- The departure vertex in the grid;
- The preceding snaxel (neighbour in clockwise direction);
- The following snaxel (neighbour in anti-clockwise direction);
- The freeze status of a snaxel.

These properties, necessary to the snaking process, are stored in a structure, allowing straight forward connectivity handling. From these, other properties can be calculated

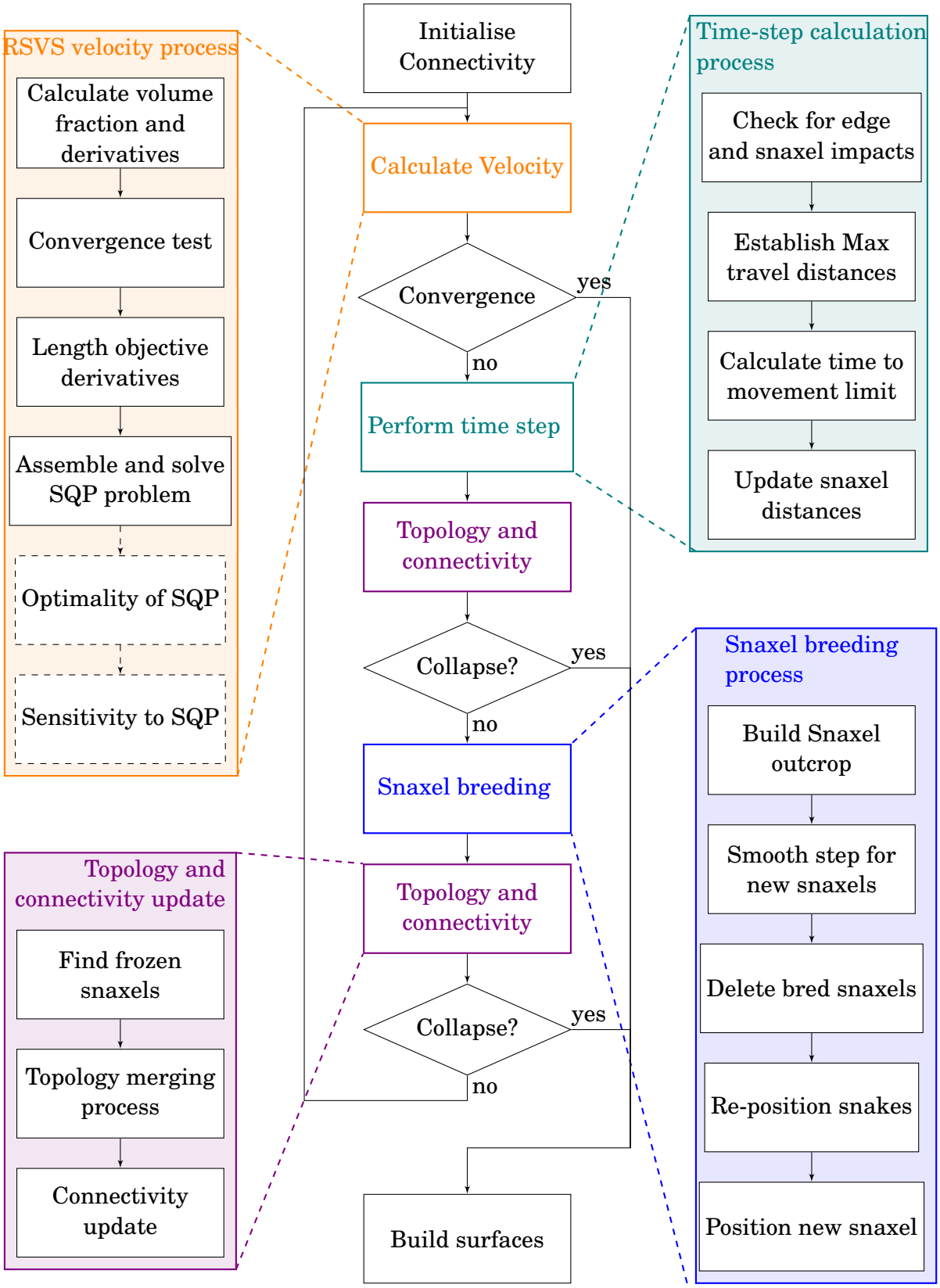


Figure 3.24: Full restricted snake algorithm for the RSVS parameterisation.

and limits on movement can be implemented. These other values can be used to evaluate the current position and update the velocity of the snake. These include:

- The snaxel position in Cartesian coordinates ($\mathbf{p}_i$);
- The direction of travel of the snaxel($\Delta\mathbf{g}_i$) and the vertex of origin ($\mathbf{g}_{\mathbf{i},\mathbf{1}}$);
- The normal vectors to the preceding and following edges ($\mathbf{n}_i$ and $\mathbf{n}_{i+1}$), $\mathbf{n}$ is defined as a specific normal corresponding to the 90 degree rotation of edge vector, i.e. an outward normal with the same length as the edge.

In general these derived properties are stored in a separate structure, which is updated after the snake is constructed at each time step. The additional properties are dependent on the application, they generally play a key role in updating the velocity and detecting the convergence of the snaking process. Only the three main properties are presented here, a larger number are evaluated during runtime to suit the needs of the program.

3.4.1.2 Position update and time step calculation

The movement of snaxels is controlled by the velocity v_i and a “time step”. While a time step is defined globally, it can also be reduced locally to avoid a self intersection or arrival at a vertex of the underlying snaking grid. Detection of profile impacts and topology changes is trivial as snaxels are constrained to move exclusively along grid lines, making it a simple comparison between two floating point values. The velocity term enables updates of d_i according to the straightforward formula in Equation 3.18.

$$d_i^{t+1} = d_i^t + \min\left(\Delta t_i, \Delta t_g\right) v_i \tag{3.18}$$

The selection of the global time step (Δt_g) depends on the method used for calculating v_i; for certain methods damping is desirable and can be introduced through the time step. Regardless of the velocity calculation method the local time step (Δt_i) is limited by the possible movement of the snaxels. Effectively snaxels cannot move beyond the edge on which they exist and are stopped at grid vertices.

Despite the term "time-step" there does not need to be a relation to a physical simulation time. The time step controls the relative step lengths of the snaxels and has two purposes: guarantee that no self-intersection happens in the r-snake, and, stabilise the numerical scheme which moves the snaxels. Limits on the time-step that maximise convergence were derived empirically from systematic benchmarking, once tuned these were not modified further.

In most cases the time step is found by calculating the pseudo-time taken by snaxels to reach a grid vertex. This condition is formulated in Equation 3.19 where $\mathbf{d}_s$ the distances of the snaxels, $\mathbf{v}_s$ the velocity of the snaxels, and exponent + and - taken to represent the positive and negative parts of those vectors respectively.

$$\begin{aligned} \Delta\mathbf{t}_+ &= \frac{1-\mathbf{d}_s}{\mathbf{v}_s^+} \text{ For positive velocities} \\ \Delta\mathbf{t}_- &= \frac{-\mathbf{d}_s}{\mathbf{v}_s^-} \text{ For negative velocities} \end{aligned} \tag{3.19}$$

In addition the time steps are used to ensure that snaxels do not intersect with each other when they exist on the same snaking grid edge. For this eventuality the time step is computed using relative velocities and distances; for snaxels i and l on the same edge:

$$\Delta t_r = \frac{d_l - d_i}{v_l - v_i} \tag{3.20}$$

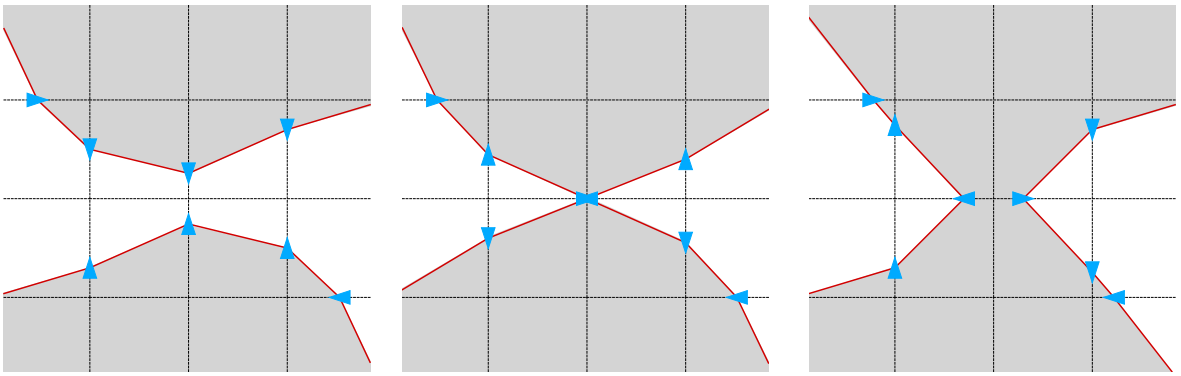
Of the three time-steps Δt_+, Δt_- and Δt_r; the smallest positive one of the three is active.

In addition to these time step requirements imposed by the snaking process, maximum time-step and step lengths are added to ensure the restricted snake process remains stable and converges. These limits, derived empirically, are similar to a Courant Friedrichs Lewy (CFL) condition. In the RSVS process these are:

- **A maximum time-step** stabilises the behaviour of the Newton algorithm avoiding large oscillations of the contour and ensures the profile stops moving when velocities decrease. This accounts for the difference between the RSVS formulation and its quadratic program (QP) approximation. 0.25 was used in this work.
- **A maximum step length** is used to avoid the repeated re-spawning of snaxels and their consequent uncontrolled propagation through the grid. This negates the impact of very large velocities that can be generated by the marching process when the velocity of a snaxel on a *snaking grid* vertex is evaluated. 0.3 was used in this work.

3.4.1.3 Snaxel breeding

While previous versions of the algorithm only allowed a single snaxel collapse per iteration [217], the current implementation allows multiple terminations at each time step through local time steps to speed up the time marching process. The process that required individual snaking steps for each snaxel is replaced by a while loop which allows



(a) Snakes before topology merging (b) Snakes at intersection (c) Snake after topology merging

Figure 3.25: Merging process for two r-snakes

snaxels to finish sequentially within a single time step. Algorithm 3 shows the process used to allow multiple snaxels to arrive and breed at a vertex on the same time step.

Once a snaxel has reached a vertex it is marked for breeding. The breeding process consists of deleting the current snaxel and creating new snaxels on all the other edges connected to that same vertex. The connectivity rules of the r-snakes were explained as follows by Bischoff and Kobbelt [231]:

- No 2 connected snaxels can be on the same edge;
- Snaxels must travel out of the profile.

This breeding process creates invalid connections which need to be removed:

- two snaxels connected by the snake which are on the same edge (one of them is removed);
- snaxel connected to two other snaxels in the same grid cell (it is removed).

This 'cleaning' process is performed iteratively until no more snaxels get removed by the process.

There is a trade-off between time accuracy and minimising iteration number; some applications require numerical stability which can only be guaranteed by close analysis of the time marching procedure. The breeding process produces a discontinuous change in the contour which leads to instability for numerical schemes where velocity is driven by information gathered at previous time steps. Unlike level sets, this process does not naturally support time accurate evolution of the contour using physical behaviours. This limitation is due to a clash between the integer nature of the connectivity changes in the r-snake and the real valued process of physics based marching.

Algorithm 3 Breeding Process

```
1: while Set of arrived snaxels is not empty do
2:     Breed first snaxel of the list
3:     Clean parametric snake of invalid connections
4:     Remove bred and deleted snaxels from the arrived set
5: end while
```

3.4.1.4 Impact, merging and freezing

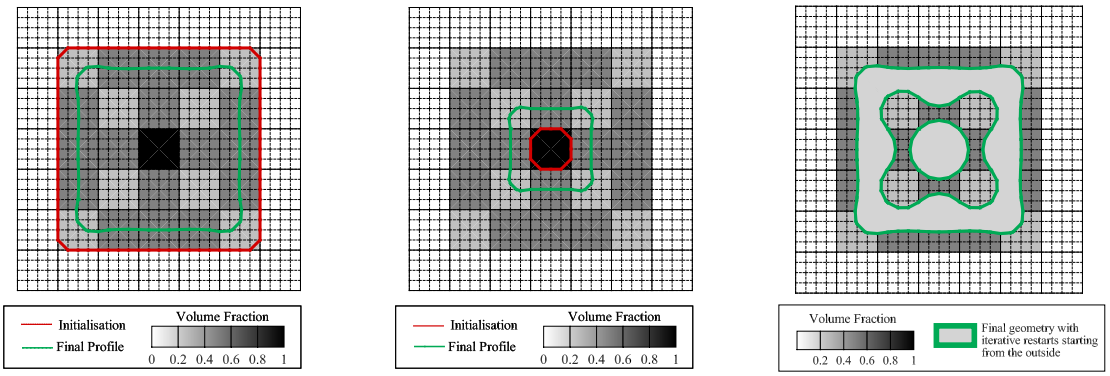
To control and evolve the topology of the r-snake, snaxel 'impacts' must be detected. By fixing snaxels on an underlying grid, r-snakes greatly simplify the detection of self-intersection: a comparison of the normalised distances of snaxels travelling on the same edge is sufficient to detect an impact. The normalised distances are compared to the expected order on the edge, which is defined when a snaxel is spawned at a vertex and then stored. This process allows arbitrarily many snaxels to exist on the same edge, in practice, when using the RSVS formulation at most two will be stable on a single snaking edge. This simple process is used to calculate the location and the time of snaxel encounters using their speed, ensuring that no self-intersection arises.

In case of an impact, topology of the contour can be either preserved or merged. If the topologies are merged the colliding snaxels are removed and the connectivity of neighbouring snaxels is altered to reflect the contact. An example of a splitting snake is shown in Figure 3.25. If topology is to be maintained the snaxels are frozen at the point of impact preventing further relative movement.

3.4.2 Snake Initialisation and Internal Voids

Initialisation of the snake is a key step in the recovery of a contour using parametric snakes. Two options have been explored: starting from a known inner boundary or from a known outer boundary. In image analysis this can be done by the user marking a point within the profile. In the volumetric parameterisation method developed in this chapter the snake is started either at the edge of empty cells or at the edge of full cells. These two types of initialisation are shown in Figure 3.26.

The choice of initialisation can have a significant impact on the final profile. Effectively the r-snakes does not guarantee that every cell will be explored and that the volume fraction will be recovered in a cell where a snaxel wasn't initialised. This issue comes from the fact that an infinity of possible snakes can recover a unique set of volume fractions. This means that the algorithm used for initialisation and snaxel stepping has



(a) Outer boundary initialisation

(b) Inner boundary initialisation

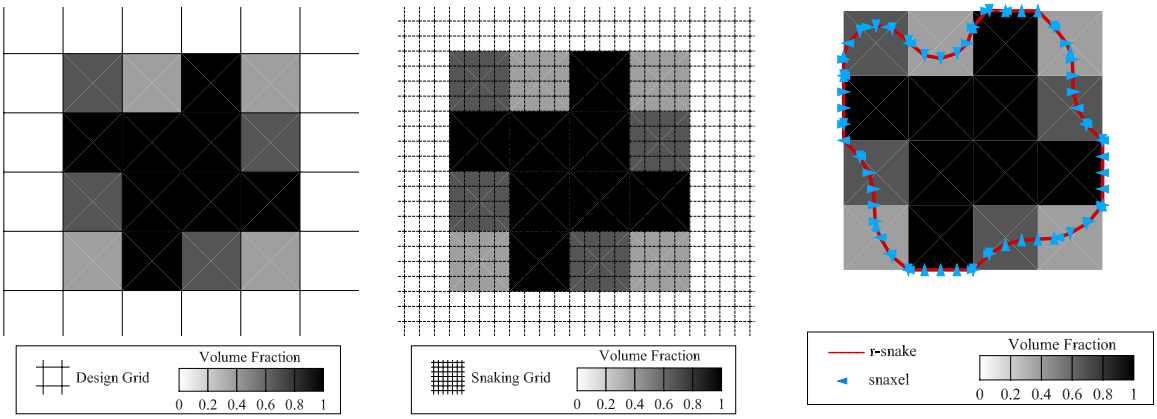
(c) Iterative outer initialisation

Figure 3.26: Types of boundary initialisations and the iterative restart process used to build internal voids in partially explored cells.

to guarantee the uniqueness of the profile recovered by the parameterisation process.

The choices at initialisation are dependent on the features desired for the output geometry. In aerodynamic design the main concern is the outer mould line, as such the external boundary is used for initialisation as shown in Figure 3.26a. Initialisation at the outer boundary is the only robust option, an internal spawn location is not defined for all values of design variables. The approach followed in the RSVS is to initialise the r-snake at the boundary between empty and non-empty parts of the design, the contour is then converged.

To ensure exploration of all cells a new restricted snake is initialised based on the remaining VOS error until all volume fractions are satisfied. This process leads to the geometry with an internal void and an internal circle of Figure 3.26c (it was also necessary to generate the bridge-like shape shown earlier in Figure 3.12). Between each iteration a snake is initialised at the outer boundaries defined by the difference between current area ($\mathbf{A}_k$) and the current target fill fraction ($\mathbf{a}_k$) in the non-explored VOS cells. For $k = 0$, the target volume fractions are the design variables; beyond they are the following difference: $\mathbf{a}_{k+1} = \mathbf{A}_k - \mathbf{a}_k$; the process stops when $\mathbf{a}_{k+1} = \mathbf{0}$. The snake generated at each iteration can be marched independently of others, or they can all be combined into a single r-snake. This second approach is preferred as it allows the various internal features to evolve and merge cohesively; allowing lower arc-length geometries and more complex topologies.



(a) Volume of Solid design grid

(b) 4 by 4 Snaking sub-grid

(c) Snake and snaxels recovering VOS information

Figure 3.27: Design grid with corresponding 4 by 4 snaking grid and an r-snake recovering the volume fractions

3.5 Marching of the R-Snake

Previous volume of solid methods [155] have shown that volume control can be an effective design variable by providing intuitive and flexible parameterisation. To achieve this the r-snake needs to be marched until a position is found that matches the volume information. However the volume condition is not sufficient to define unique and smooth profiles with continuous changes in geometry which were identified as desirable features of aerodynamic parameterisation methods. These properties are achieved in the RSVS using length minimisation, which is analogous to an internal tensile force applied to the profile tension was preferable to other terms (curvature, analogous to bending) as it allows for smooth profiles in most cases but can also recover sharp corners where the VOS requires it. These other objective functions would enable different properties: minimising curvature would probably lead to profiles with higher order derivative continuity (G_2).

To allow a high degree of geometric flexibility with few design variables, features need to be recovered below the resolution of the VOS design grid. To build a parameterisation method the r-snake is integrated with a grid on which volume information is specified (Figure 3.27a). To make the active contour compatible with the VOS formulation, the r-snake is marched on a shape conserving refinement of the VOS grid (Figure 3.27b).

The volume condition of the design variable and the length minimisation need to be integrated in order to form a robust method to march the r-snake. The use of a physical time-marching procedure was not seen as appropriate for the r-snake as the *breeding*

process described in Section 3.4.1.3 introduces a discontinuity and presents a significant challenge to a physical system. Consequently the marching procedure is framed as a mathematical programming problem of minimising the length of the r-snake under a set of volume constraints. The continuous form of the program was presented in Section 3.2.1 Equation 3.1; the discrete form of the equation that is applied to the r-snake is repeated below (Equation 3.21).

$$\begin{aligned} &\min_{\mathbf{d}} \sum_{i=1}^{n} |\mathbf{p}_i - \mathbf{p}_{i-1}| \quad \text{with } \mathbf{p}_i = \Delta \mathbf{g_i} d_i + \mathbf{g}_{i,1} \\ &\text{s.t. } \mathbf{A(d)} - \mathbf{a} = \mathbf{0} \end{aligned} \tag{3.21}$$

In Equation 3.21 $\mathbf{d}$ is the column vector of all snaxel distances d_i, $\mathbf{a}$ is the column vector of target volumes in each VOS cell and $\mathbf{A(d)}$ is the current volume in each VOS cell contained by the r-snake. The normalised snaxel distances $\mathbf{d}$ are the design variables of the RSVS parameterisation length minimisation. The following sub-sections show how this problem can be solved efficiently by using a Newton step sequential quadratic programming (SQP) procedure. The availability of analytical gradients for the profile length condition (the objective function) and the volume information (constraints) means an efficient gradient based method may be applied.

3.5.1 Derivation of the SQP for Snake Marching

To solve this optimisation problem a method was required that would require few iterations and function evaluations to converge. For this reason the obvious choice is to use a gradient based method. The availability of analytical first and second derivatives means that SQP is a viable option. A damped Newton step defined from a quadratic approximation to the full mathematical program is used to advance the solution. The Newton step does away with the line search, ensuring that few snaking steps are required. Validation of the damped Newton-SQP step process for a single volume design variable was presented in Figure 3.10a in Section 3.3. The Newton step SQP equations presented below are derived in Boggs and Tolle [49] and were implemented by the authors into the snaking process to calculate the velocity of the snaxel v_i.

The first step to using an SQP algorithm is to formulate the Lagrangian of the optimisation problem; the Lagrangian is presented in Equation 3.24. This equation is best rewritten into its standard formulation in terms of the objective function (f), the constraints ($\mathbf{h}$), presented respectively in Equations 3.22 and 3.23; and the Lagrange multipliers ($\boldsymbol{\lambda}$).

$$f(\mathbf{d}) = \sum_{i=1}^{n} |\mathbf{p}_i - \mathbf{p}_{i-1}| \tag{3.22}$$

$$\mathbf{h}(\mathbf{d}, \mathbf{a}) = [\mathbf{A}(\mathbf{d})] - \mathbf{a} \tag{3.23}$$

$$\mathscr{L}(\mathbf{d}, \boldsymbol{\lambda}, \mathbf{a}) = \sum_{i=1}^{n} |\mathbf{p}_i - \mathbf{p}_{i-1}| + \sum_{j=1}^{m} \lambda_j \left(A_j(\mathbf{d}) - a_j\right) \tag{3.24}$$

SQP algorithms progress by approximating the full non-linear program (NLP) as a quadratic sub-problem at every step k. This sub-problem can be solved using any method suitable for quadratic optimisation. The classical approximating QP developed by Boggs and Tolle [233] uses the Hessian of the Lagrangian ($\mathbf{H}^k_{\mathbf{d}}\mathscr{L}^k$) in its formulation; however many implementations rely on approximations to tailor the convergence behaviour of the optimisation. In this case, the sub-problem is solved using Newton's method for which the expression is presented in Equation 3.16 and repeated below, and derived in [49]. The algorithm used here differs only through the use of the Hessian of the objective ($\mathbf{H}_{\mathbf{d}}f$) rather than that of the Lagrangian ($\mathbf{H}_{\mathbf{d}}\mathscr{L}$). This is because the Hessian of the constraints is often poorly conditioned and can lead to instability. The Hessian of the area constraint can be added in close to the optima and can significantly accelerates convergence.

$$\boldsymbol{\lambda}^{k+1} = \left((\nabla_{\mathbf{d}}\mathbf{h})^T (\mathbf{H}_{\mathbf{d}}f)^{-1} (\nabla_{\mathbf{d}}f)\right)^{-1} \left(\mathbf{h} - (\nabla_{\mathbf{d}}\mathbf{h})^T (\mathbf{H}_{\mathbf{d}}f)^{-1} (\nabla_{\mathbf{d}}f)\right)$$
$$\mathbf{d}^{k+1} - \mathbf{d}^k = -(\mathbf{H}_{\mathbf{d}}f)^{-1} \left((\nabla_{\mathbf{d}}f) + (\nabla_{\mathbf{d}}\mathbf{h})\boldsymbol{\lambda}^{k+1}\right)$$

The change in design variables $\Delta^{k+1}_{\mathbf{d}}$ is used as the velocities (v_i) of the snaxels with the final step chosen by the snaking process. To implement the SQP of Equation 3.16, the following derivatives must be computed: the Jacobian of the constraints ($\nabla_{\mathbf{d}}\mathbf{h}$); the gradient of the objective ($\nabla_{\mathbf{d}}f$) and the Hessian of the objective ($\mathbf{H}_{\mathbf{d}}f$). Thanks to the formulation of the snaking process all these values are available analytically by differentiating the appropriate area and snaxel position with respect to the distances d_i.

3.5.2 Measure of RSVS Convergence

Termination of the RSVS process requires a measure of the convergence of the profile, and its distance to the length minimising r-snake. A number of measures of convergence are used all of which are Root Mean Square (RMS) errors, where the RMS of a set of n general error terms x_i is shown in Equation 3.25. Optimality (e_{opt} Equation 3.26) and feasibility (e_{feas} Equation 3.27) of the SQP are valuable measures to indicate convergence of the

RSVS. They are the root mean squared (RMS) error of the gradient and constraint violation respectively.

On VOS grids with cells of different sizes, these metrics do not always perfectly reflect the desired convergence on small features of a profile. To ensure reliable representation and smooth evolution of small geometric features it is desirable to attribute all snaxels and VOS cells the same importance in the convergence metric. Instead of optimality and feasibility, the RMS of the volume fraction error (e_{vf} Equation 3.28) and of the snaxel velocities (e_{vel} Equation 3.29) is used. These metrics naturally normalise the size of the errors leading to the desired behaviour of the convergence metric.

$$RMS(\mathbf{x}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i)^2} \tag{3.25}$$

$$e_{opt} = RMS\left((\nabla_{\mathbf{d}} f) + (\nabla_{\mathbf{d}} \mathbf{h})\,\boldsymbol{\lambda}\right) \tag{3.26}$$

$$e_{feas} = RMS(\mathbf{h}) \tag{3.27}$$

$$e_{vf} = RMS(\mathbf{A}(\mathbf{d}) - \mathbf{a}) \tag{3.28}$$

$$e_{vel} = RMS(\mathbf{v}) \tag{3.29}$$

3.5.3 Calculation and Differentiation of the Volume Fraction

Calculation of the volume fraction in two dimensions is performed using Green's theorem in the plane.

$$\iint_D \left(\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y}\right) dxdy = \oint_C (Ldx + Mdy) \tag{3.30}$$

This is related to both the three dimensional divergence theorem and Stoke's theorem. Then the area in 2D, $dAc = dxdy$, (generically called the volume in this work) is calculated by choosing L and M such that $\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y} = 1$.

$$A_C = \iint_D dA = \oint_C (Ldx + Mdy) \tag{3.31}$$

$$A_C = \oint_C xdy = -\oint_C ydx = \frac{1}{2}\oint_C (xdy - ydx) \tag{3.32}$$

The two dimensional form of the theorem for a polygon is presented in Equation 3.33. As this is a special case of the divergence theorem, similar results in higher dimensions are available, making feasible a 3 dimensional adaptation of this method. This discrete

form rests on two values: $\mathbf{q}$ the coordinate of centre of the edge of polygons, and, $\mathbf{n}$ the outward facing normal defined by the 90 degree rotation of the edge vector.

$$A_C = \frac{1}{2}\sum_C \mathbf{q}\cdot\mathbf{n} \quad \text{with}: \mathbf{q} = \frac{\mathbf{p}_i + \mathbf{p}_{i+1}}{2} \text{ and } \mathbf{n} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (\mathbf{p}_i - \mathbf{p}_{i+1}) \tag{3.33}$$

This equation is used within each cell to define the area currently contained by the snakes. Manipulation of Equation 3.33 allows it to be formulated into the matrix product in Equation 3.34. Vector $\mathbf{p}$ is the list of coordinates of the vertices where $\mathbf{p}_n$ is the coordinate row vector of the nth vertex defining polygon C. Equation 3.34 is derived from the decomposition of $\mathbf{q}$ and $\mathbf{n}$ into, respectively, the mean and the difference of neighbouring vertices which is readily transformed into simple matrix equations that can be assembled to reflect the connectivity information of a polygon.

$$A_C = \frac{1}{2}\sum_C^i \left(\frac{1}{2}(\mathbf{p}_i + \mathbf{p}_{i-1})^T \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} (\mathbf{p}_i - \mathbf{p}_{i-1}) \right) = \frac{1}{4}\mathbf{p}^T \mathbf{R}_A \mathbf{p} \tag{3.34}$$
$$\text{where}: \quad \mathbf{p} = \left\{ \mathbf{p}_1,\ \mathbf{p}_2,\ \cdots,\ \mathbf{p}_n \right\}^T$$

Matrix $\mathbf{R}_A$ (Equation 3.35, Figure 3.28) is the result of the decomposition of the centre point and the normal of a segment into vector equations. Assembling it as a single matrix equation is a convenient way of expressing the first two derivatives without somewhat obscure index notations.

$$\mathbf{R}_A = \begin{bmatrix}
0 & 0 & 0 & 1 & & \dots & 0 & \dots & & 0 & -1 \\
0 & 0 & -1 & 0 & & \dots & 0 & \dots & & 1 & 0 \\
0 & -1 & 0 & 0 & 0 & 1 & & & & & \\
1 & 0 & 0 & 0 & -1 & 0 & \ddots & & & & \\
 & & 0 & -1 & 0 & 0 & & & & & \\
\vdots & \vdots & 1 & 0 & 0 & 0 & & & & & \\
0 & 0 & & \ddots & & & \ddots & & & \ddots & \\
\vdots & \vdots & & & & & & 0 & 0 & 0 & 1 \\
 & & & & & & & 0 & 0 & -1 & 0 \\
0 & 1 & & \dots & 0 & \dots & \ddots & 0 & -1 & 0 & 0 \\
-1 & 0 & & \dots & 0 & \dots & & 1 & 0 & 0 & 0
\end{bmatrix} \tag{3.35}$$

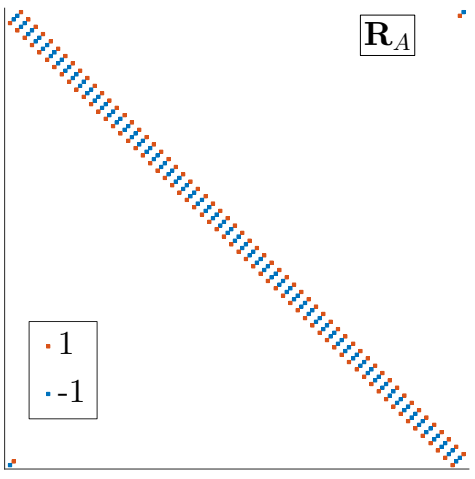


Figure 3.28: Matrix $\mathbf{R}_A$ for the 62 snaxel profile of Figure 3.9.

The SQP algorithm being developed requires the derivation of the Jacobian of the area constraints with respect to the non-dimensional snaxel distances. Applying the matrix form of Green's theorem to the VOS cells greatly simplifies the calculation of the derivatives as all the connectivity information is precomputed and hidden into matrix $\mathbf{R}_A$. For the derivative to be computed, vector $\mathbf{p}$ is readily separated into a variable and constant part. Recalling the formulation of $\mathbf{p}_i$ (Equation 3.21) and of $\mathbf{p}$ (Equation 3.34).

$$\mathbf{p} = \Delta\mathbf{G}\,\mathbf{d}_C + \mathbf{g}_1 \qquad \text{where} \quad \begin{cases} \Delta\mathbf{G} = \begin{bmatrix} \Delta\mathbf{g}_{1,x} & \Delta\mathbf{g}_{1,y} & 0 & 0 & 0 \\ 0 & 0 & \Delta\mathbf{g}_{2,x} & \Delta\mathbf{g}_{2,y} & 0 \\ 0 & 0 & 0 & 0 & \ddots \end{bmatrix}^T \\ \mathbf{d}_C = \{d_1,\, d_2,\, \cdots,\, d_n\}^T \\ \mathbf{g}_1 = \left\{\mathbf{g}_{1,1},\, \mathbf{g}_{2,1},\, \cdots,\, \mathbf{g}_{n,1}\right\}^T \end{cases} \tag{3.36}$$

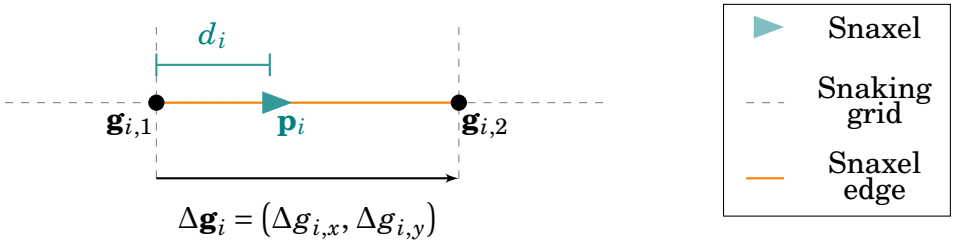


Figure 3.29: Graphical representation of the quantities used in the area calculation and differentiation.

This formulation shows that p is only a function of the distance along edges of the snaxels (d_i) and properties of the snaxel grid, the direction of a travelling snaxel ($\Delta\mathbf{g}_i$) and the originating grid point ($\mathbf{g}_{i,1}$); all are represented in Figure 3.29. For any vertex of the polygon which is not a snaxel the entry into $\mathbf{d}_C$ is replaced by a 0. Replacing Equation 3.36 into Equation 3.34 Equation 3.37 is developed. This form of the equation simplifies the differentiation process with regard to $\mathbf{d}$.

$$A_C = \frac{1}{4}\left[\left(\mathbf{d}_C^T\,\Delta\mathbf{G}^T\right)[\mathbf{R_A}](\Delta\mathbf{G}\,\mathbf{d}_C) + 2\mathbf{g}_1^T\,[\mathbf{R_A}](\Delta\mathbf{G}\,\mathbf{d}_C) + \mathbf{g}_1^T\,[\mathbf{R_A}]\,\mathbf{g}_1\right] \tag{3.37}$$

$$\nabla_{\mathbf{d}} A_C = \frac{1}{2}\mathbf{I_d}\Delta\mathbf{G}^T\,[\mathbf{R_A}]\,\Delta\mathbf{G}\,\mathbf{d}_C + \frac{1}{2}\mathbf{g}_1^T\,[\mathbf{R_A}]\,\Delta\mathbf{G}\,\mathbf{I_d} \tag{3.38}$$

$$\mathbf{H_d} A_C = \frac{1}{2}\mathbf{I_d}\Delta\mathbf{G}^T\,[\mathbf{R_A}]\,\Delta\mathbf{G}\mathbf{I_d} \tag{3.39}$$

The differentiation relies on simple matrix derivation rules and the symmetric nature of $[\mathbf{R_A}]$. In the previous equations $\mathbf{I_d}$ is the result of the operation $\nabla_{\mathbf{d}}\mathbf{d}_C$; it is a

rectangular matrix of ones and zeros which has the effect of deleting rows and columns from the equations corresponding to static vertices and inactive snaxels. $\nabla_{\mathbf{d}} A_C$ is the gradient of a VOS cell; it is a column vector of length n (the number of snaxels in the profile). To build the full Jacobian of the constraint ($\nabla_{\mathbf{d}} \mathbf{h}$) the gradient in each VOS cell is calculated using Equation 3.38 and the resulting vectors are assembled to form the matrix of Equation 3.40.

$$\nabla_{\mathbf{d}} \mathbf{h} = \left[\nabla_{\mathbf{d}} A_1,\ \nabla_{\mathbf{d}} A_2,\ \cdots,\ \nabla_{\mathbf{d}} A_j,\ \cdots,\ \nabla_{\mathbf{d}} A_m \right] \tag{3.40}$$

3.5.4 Differentiation of the Objective Function

The SQP algorithm described in Equation 3.21 requires the gradient ($\nabla_{\mathbf{d}} f$) and the Hessian ($H_{\mathbf{d}} f$) of the objective function with regard to the design variable vector $\mathbf{d}$. These can be calculated for each segment of the snake. The Euclidean norm of the edge vector can be decomposed into Equations 3.41 and 3.42. Using the same notation as Equation 3.21 and defining $\mathbf{p}_i - \mathbf{p}_{i-1} = \mathbf{F}_i$.

$$|\mathbf{F}_i| = \sqrt{\alpha_i d_i^2 + \alpha_{i-1} d_{i-1}^2 + \alpha_{i,i-1} d_i d_{i-1} + \beta_i d_i + \beta_{i-1} d_{i-1} + c + \alpha_i \alpha_{i-1} \varepsilon^2} \tag{3.41}$$

$$\begin{aligned} \text{with}: \quad \alpha_i &= \Delta\mathbf{g}_i \cdot \Delta\mathbf{g}_i & \beta_i &= 2\Delta\mathbf{g}_i \cdot \left(\mathbf{g}_{i,1} - \mathbf{g}_{i-1,1}\right) \\ \alpha_{i-1} &= \Delta\mathbf{g}_{i-1} \cdot \Delta\mathbf{g}_{i-1} & \beta_{i-1} &= -2\Delta\mathbf{g}_{i-1} \cdot \left(\mathbf{g}_{i,1} - \mathbf{g}_{i-1,1}\right) \\ \alpha_{i,i-1} &= -2\Delta\mathbf{g}_i \cdot \Delta\mathbf{g}_{i-1} & c &= \left(\mathbf{g}_{i,1} - \mathbf{g}_{i-1,1}\right) \cdot \left(\mathbf{g}_{i,1} - \mathbf{g}_{i-1,1}\right) \end{aligned} \tag{3.42}$$

This form allows a much more readable representation of the first and second derivatives of the function. ε is a small positive number used to stabilise the derivatives as the distance goes to zero. Considering the differentiation of 3.41 with regard to d_γ an arbitrary component of $\mathbf{d}$, three cases are identified.

$$\begin{aligned} \gamma \neq i \text{ and } \gamma \neq i-1 \text{ then}: \quad & \frac{\partial |\mathbf{F}_i|}{\partial d_\gamma} = 0 \\ \gamma = i \text{ then}: \quad & \frac{\partial |\mathbf{F}_i|}{\partial d_i} = \frac{\left(2\alpha_i d_i + \alpha_{i,i-1} d_{i-1} + \beta_i\right)}{2|\mathbf{F}_i|} \\ \gamma = i-1 \text{ then}: \quad & \frac{\partial |\mathbf{F}_i|}{\partial d_{i-1}} = \frac{\left(2\alpha_{i-1} d_{i-1} + \alpha_{i,i-1} d_i + \beta_{i-1}\right)}{2|\mathbf{F}_i|} \end{aligned} \tag{3.43}$$

A similar process is followed for the second differences where the differentiation is carried out with respect to d_γ and d_α.

$$
\begin{aligned}
\gamma \text{ or } \alpha \neq \{i, i-1\} \text{ then}: \\
\frac{\partial^2 |\mathbf{F}_i|}{\partial d_\gamma \partial d_\alpha} &= 0 \\
\gamma = \alpha = i \text{ then}: \\
\frac{\partial^2 |\mathbf{F}_i|}{\partial d_i^2} &= \frac{4\alpha_i |\mathbf{F}_i|^2 - \left(2\alpha_i d_i + \alpha_{i,i-1} d_{i-1} + \beta_i\right)^2}{4|\mathbf{F}_i|^3} \\
\gamma = \alpha = i-1 \text{ then}: \\
\frac{\partial^2 |\mathbf{F}_i|}{\partial d_{i-1}^2} &= \frac{4\alpha_{i-1} |\mathbf{F}_i|^2 - \left(2\alpha_{i-1} d_{i-1} + \alpha_{i,i-1} d_i + \beta_{i-1}\right)^2}{4|\mathbf{F}_i|^3} \\
\gamma = i \ ; \ \alpha = i-1 \text{ then}: \\
\frac{\partial^2 |\mathbf{F}_i|}{\partial d_i \partial d_{i-1}} &= \frac{\begin{array}{r} 2\alpha_{i,i-1} |\mathbf{F}_i|^2 - \left(2\alpha_i d_i + \alpha_{i,i-1} d_{i-1} + \beta_i\right) \\ \times \left(2\alpha_{i-1} d_{i-1} + \alpha_{i,i-1} d_i + \beta_{i-1}\right) \end{array}}{4|\mathbf{F}_i|^3}
\end{aligned}
\tag{3.44}
$$

The value of ε (Equation 3.41) is chosen to ensure that the denominator of the derivatives does not go to 0. A value is selected such that the impact on the derivative is limited to a small region around singularities. Figures 3.30 and 3.31 show the derivatives for $\varepsilon = 0$ and $\varepsilon = 10^{-6}$ respectively, the singularity in the first and second derivative as d_i and $d_i - 1$ tends to zero highlight the need for this parameter. The effect of this term is to create a saddle point close to grid vertices, this means that snaxels which arrive at corners and are not redistributed by the snaking process can get stuck. These saddle points can appear to the SQP as narrow local minima; however these are avoided using small values of ε and aggressively stepping through vertices of the snaking grid. Empirical investigations indicate that ε should be an order of magnitude smaller than the *arrival trust region* of the snaking process. The value of ε (Equation 3.41) is chosen to ensure that the denominator of the derivatives does not go to 0 and is sufficiently high to ensure good conditioning of the Hessian.

Equations 3.43 and 3.44 can be summed over all snaxels to build the Jacobian and the Hessian of the objective function. The Hessian is a tridiagonal symmetric matrix (Equation 3.44, seen in Figure 3.32) which means the cost of inverting it for the calculation of the Newton step (Equation 3.16) is low. This is due to the formulation of the tensile force, as it only relies on one neighbour on each side it leads to a sparse Hessian which favours the stability of the system. A value for ε is selected such that the

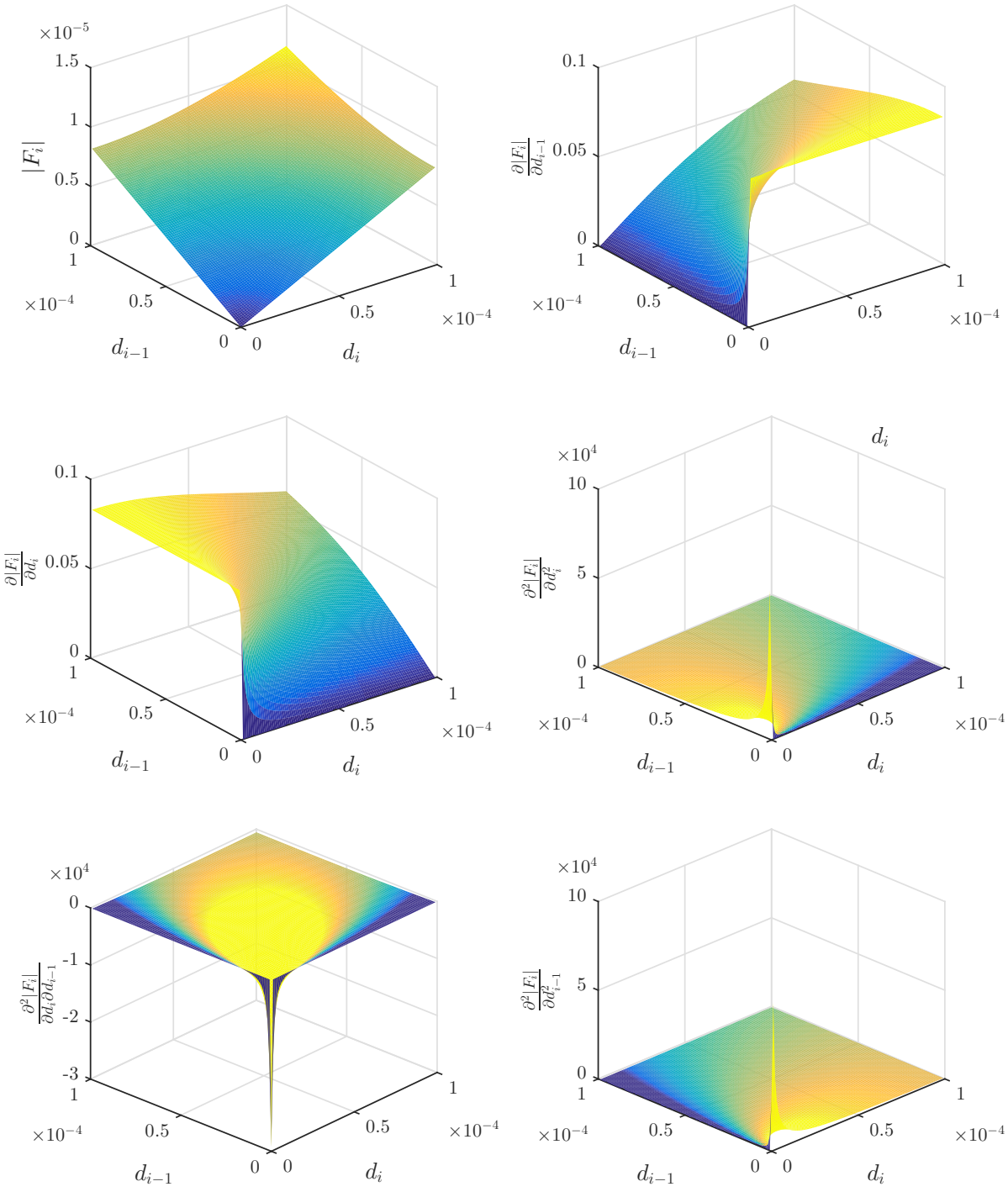


Figure 3.30: Objective function value, 1st derivatives and 2nd derivatives for two snaxels converging on a vertex, $\varepsilon = 0$

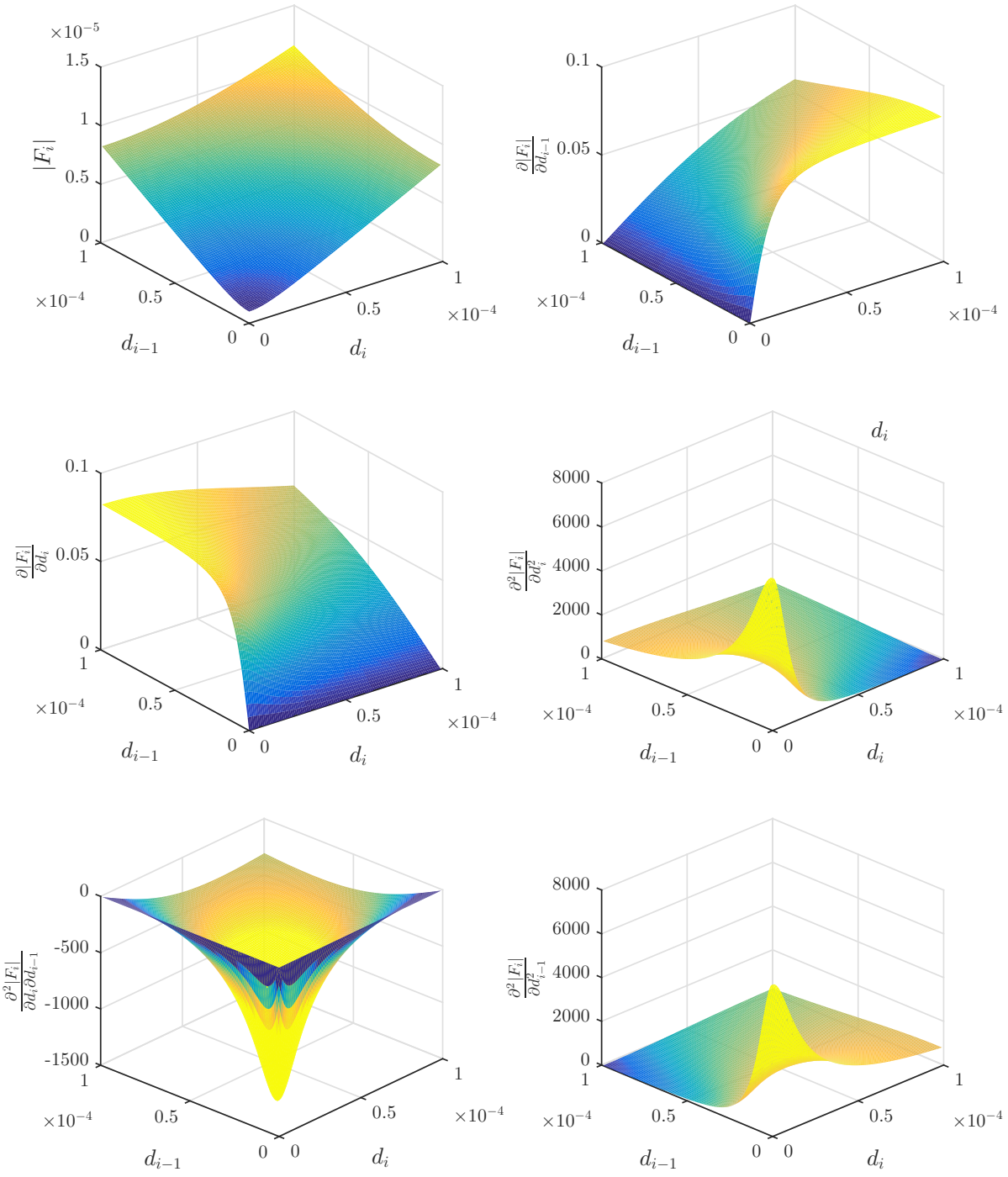


Figure 3.31: Objective function value, 1st derivatives and 2nd derivatives for two snaxels converging on a vertex, $\varepsilon = 10^{-6}$

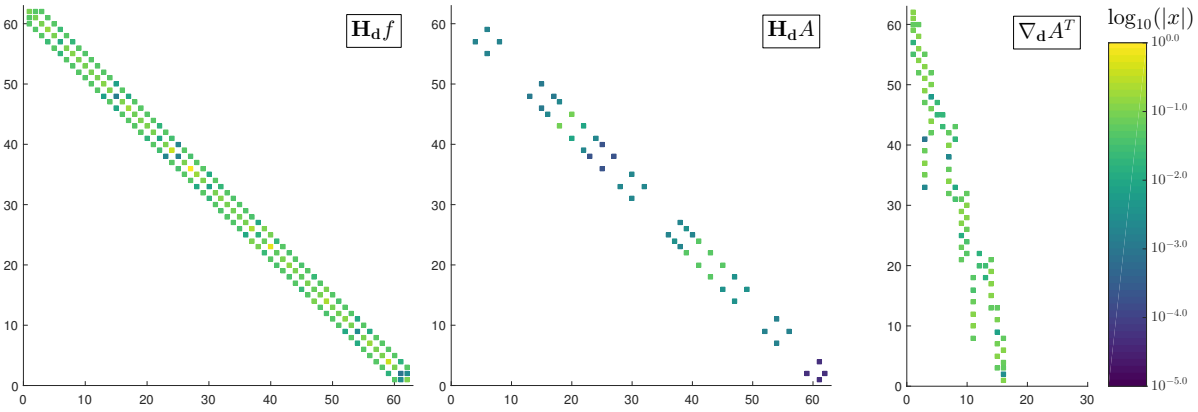


Figure 3.32: Sparsity plots of RSVS derivatives for a normally parameterised profile.

impact on the derivative is limited to a small region around singularities. A typical value for this parameter will be of the order 10^{-5}.

3.6 Validation of the Analytical Shape of the RSVS

Section 3.2.2 showed that the continuous RSVS governing minimisation problem could be solved analytically using calculus of variations. That section showed that the solution to the RSVS problem is a piecewise continuous curve made of arcs of circle. The parameterisation results shown in Section 3.3 do certainly look smooth, and, small profiles seem to tend to circular solutions; however a quantitative validation of this behaviour is needed.

While the analytical representation derived in section 3.2.2.1 is unwieldy, arcs of circles can be represented with non-uniform rational B-Splines (NURBS). In this section, this equivalence is exploited to define an analytical representation of a RSVS solution, this solution is then compared to results of the r-snake iterative process. This comparison validates the behaviour of the discrete geometry generation method showing that it is correctly minimising lengths under area constraints.

3.6.1 Calculation of the Equivalent NURBS Patches

Non-uniform rational B-Splines (NURBS) are a generalisation of B-Spline curves for which points can be non-uniformly weighted. This type of curve is suitable for computational applications as they can be stored with very little memory, and evaluated by fast and stable algorithms. In addition, the relationship between the control points and the final curves are intuitive, explaining why NURBS form the core of many computational

aided design (CAD) packages. NURBS are defined by: their degree, weighted control points and a knot vector. The order of the curve controls the number of points acting in each region of the curve and the polynomial degree (d) of the basis of the NURBS: a curve of *order* 3 is of *degree* 2 and will have parabolic basis functions. The control points ($\mathbf{P}_i$) are a set of ordered points which can be open or closed. These control points are associated to an equal number of weights (w_i) which control relative impact of each point. The NURBS curve is evaluated along non-dimensional parameter u. The knot vector (k_i) defines the region of influence of each basis function and control point. The knot vector is of the size of the number of points plus the order of the NURBS curve. Equations 3.45 to 3.47 present the Bezier, recursive, formulation of general NURBS. For more details of implementation and applications the reader is referred to the book by Piegl and Tiller [234].

$$N_{i,d} = f_{i,d}N_{i,d-1} + g_{i+1,d}N_{i+1,d-1} \qquad N_{i,0}(u) = \begin{cases} 1 & \text{for}: k_i < u \leq k_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{3.45}$$

$$f_{i,d}(u) = \frac{u - k_i}{k_{i+d} - k_i} \qquad g_{i,d}(u) = 1 - f_{i,d}(u) = \frac{k_{i+d} - u}{k_{i+d} - k_i} \tag{3.46}$$

$$C(u) = \sum_{i=1}^{k} R_{i,d}(u)\mathbf{P}_i \qquad R_{i,d}(u) = \frac{N_{i,d}(u)w_i}{\sum_{j=1}^{k} N_{j,d}(u)w_j} \tag{3.47}$$

The ability of NURBS to represent conic sections exactly including arcs of circles is well documented, and can be done in many different ways. Grothmann and Sommer [235] investigated the minimal NURBS required to generate a given set of patched circular arcs. The minimal NURBS to represent a patched conic is of order 3 (second order polynomials); and has 1 control point at each end of the spline and 1 control point per circular patch. The control points are placed at the intersection of the tangents taken at the start and end of each circular patches; weights and knot positions are then computed to tangency and circularity. The system to solve each circular spline pattern derived in this way requires the solution of a stiff system; to build the RSVS NURBS equivalents additional control points are added at the tangency points. Figure 3.33 shows the difference between the minimal and the RSVS equivalent NURBS layout. The weights for the layout used for the RSVS are 1 at each tangential point and chosen to match the area in the between.

To build a NURBS geometry equivalent to the RSVS profiles the definitions of Figure 3.34 are used. Using Equations 3.48 to 3.51, the steps listed below define the knots, control points and weights of NURBS patches equivalent to the RSVS:

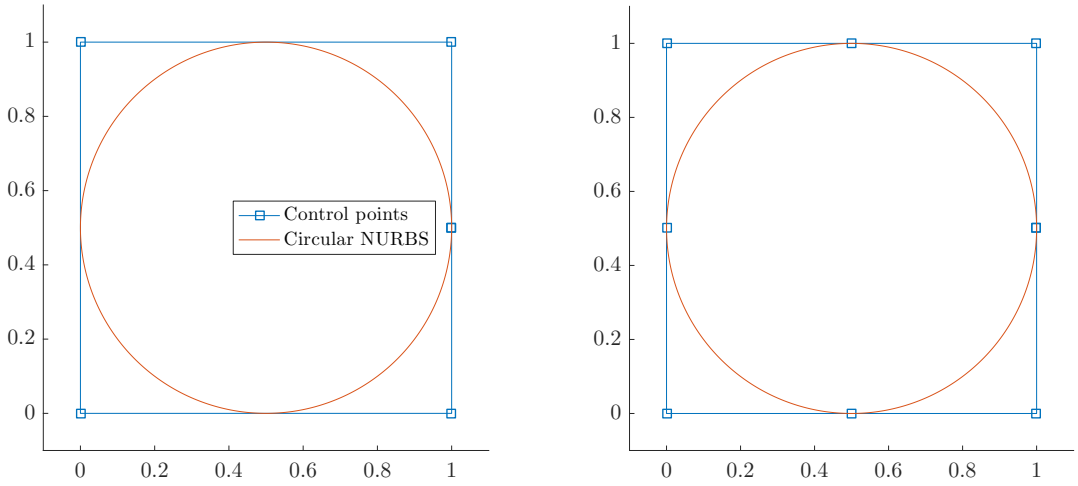


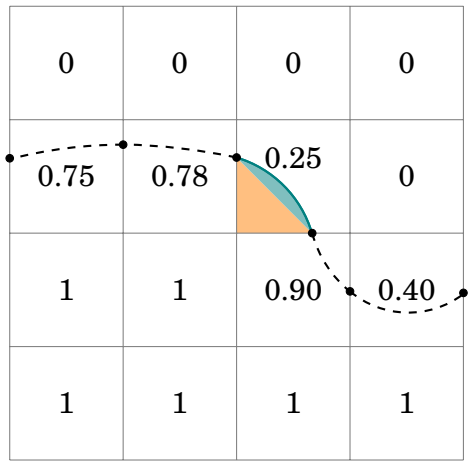
Figure 3.33: Minimal NURBS for a full circle (left) and, NURBS representation chosen to translate the RSVS (right).

1. Keep the snaxels lying on boundaries between VOS cells and use them as the 'tangential' control points;
2. Define control point $P_1 = N_0 + h(P_2 - P_0)^{\perp}$;
3. Tune h using golden section search until the value of $A_{arc} + A_{poly}$ is equal to the required volume fraction;
4. With $w_0 = w_2 = 1$ then $w_1 = cos(\beta) = sin(\gamma) = |P_0 - N_0|/|P_0 - P_1|$;
5. The knot vector is $\{0, 0, 0, 1/M, 1/M, ..., j/M, j/M, ..., 1, 1, 1\}$ $\forall j \in \{0, ..., M-1\}$ where M is the number of NURBS patches.

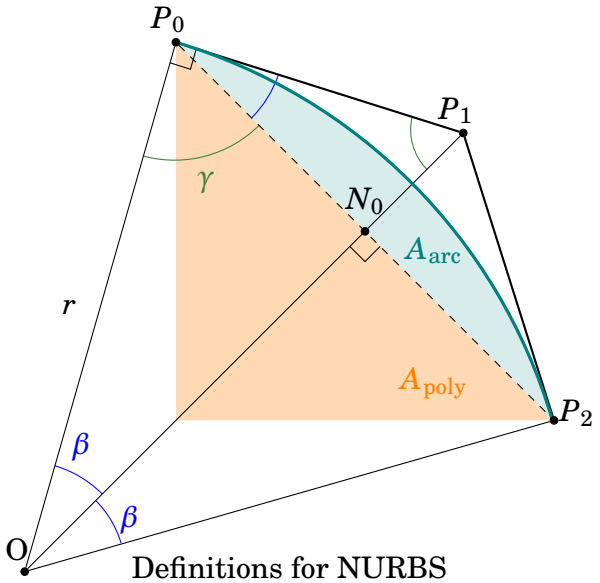
The NURBS generated by this process is not an exact solution to the RSVS governing equation: the derivative condition is not imposed. Indeed, the control points with unit weight and the repeated values in the knot vector lead to a discontinuity in gradient of the profile if the following control points are not aligned. In order for the gradient condition to be achieved an additional iteration over the position of the tangential points would need to be performed. This exact solution to the length minimisation area constraint problem using NURBS is in essence an alternate approach to this volumetric parameterisation and was not undertaken. There would not have been much benefit in terms of the optimisation framework used in later chapters of this thesis.

3.6.2 Validation of the NURBS Equivalence

In this section the mathematics developed in the previous sections are validated numerically. This validation process includes an example of the NURBS equivalence with a



VOS grid and RSVS profile



Definitions for NURBS patch calculation

Figure 3.34: Definition of a NURBS from the RSVS profile.

$$\sin(\gamma) = \frac{|P_0 - N_0|}{|P_0 - P_1|} = \frac{|N_0 - O|}{r} \qquad \cos(\gamma) = \frac{|P_1 - N_0|}{|P_0 - P_1|} = \frac{|P_0 - N_0|}{r} \tag{3.48}$$

$$\tan(\gamma) = \frac{|N_0 - O|}{|P_0 - N_0|} = \frac{|P_0 - N_0|}{|N_0 - P_1|} \qquad \beta = \frac{\pi}{2} - \gamma \tag{3.49}$$

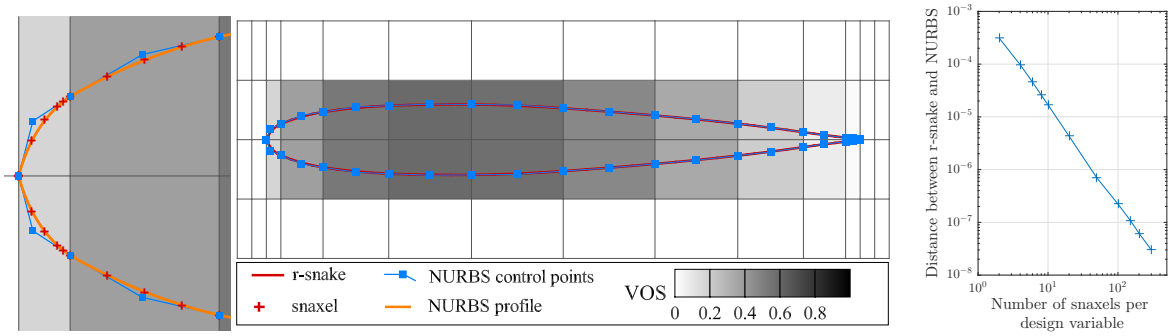
$$r = \frac{|P_0 - P_1|\,|P_0 - N_0|}{|P_1 - N_0|} \tag{3.50}$$

$$A_{\text{arc}} = \beta r^2 - |O - N_0|\,|N_0 - P_0| = \beta \frac{|P_0 - P_1|^2\,|P_0 - N_0|^2}{|P_1 - N_0|^2} - \frac{|P_0 - N_0|^3}{|N_0 - P_1|} \tag{3.51}$$

profile generated with the current implementation of the RSVS.

The key consideration in allowing good performance of the algorithm is the relation between the design grid (carrying the VOS information) and the snaking grid (over which the r-snake evolves). As was discussed in Section 3.3 the coarseness of the snaking grid has a significant impact on the stability and convergence of the RSVS. For most applications a cell refinement level of 4 (each volume cell is split into 16) was found to produce quick convergence. However the resolution of the snaking grid also controls the convergence of the RSVS on the piecewise circular arcs.

Due to the discrete nature of the r-snake, the profiles generated by the RSVS only approach the shape predicted by the analytical calculations. As the number of snaxels



(a) NACA 0012 profile used for NURBS validation, including r-snake and NURBS control points.

(b) Convergence on NURBS definition.

Figure 3.35: Change in normal distance between analytically derived NURBS representation and r-snake with increasing snaxel density.

per VOS cell is increased by using finer snaking grids the distance between the circular patches generated by NURBS and the r-snake converge. The convergence of this error is shown for the representation of a NACA 0012 airfoil going from a very coarse snaking grid of 2 snaxels per design variables to a finest grid of 300 snaxels per design variable in Figure 3.35. The NURBS representation is built by keeping the snaxels that lie on the edges of the design grid. These are then linked by a circular NURBS patch which satisfies the VOS requirement in the cell.

One of the key benefits of this parameterisation method over previous VOS methods is its natural ability to build sharp corners. As the required volume fraction is decreased at the edge of the profile, the minimisation of the profile length tends to create a very small feature which tends to a sharp corner as the volume fraction tends to 0. This effect can be seen in Figure 3.35, where both the leading and trailing edge are fixed in place by VOS values of 10^{-5} at the extremeties. This use of small volume fractions to modify the properties of the curves is analogous to the introduction of a knot inside a spline. A smoother leading edge can be achieved by removing these volume fractions or by designing grids with more control at the leading edge as in Figure 3.23. This ability to transition between sharp and smooth shapes is very important to the design of useful aerodynamic bodies which often require sharp trailing edges or leading edges. These small volume fractions can also be used to fix the length of a profile by effectively pinning leading and trailing edge position.

3.6.3 Implications of the Existence of Analytical Limit Curves

While the RSVS formulation defines geometries implicitly, which can limit access to the final geometry for interface with other methods; the RSVS is generated using an explicit representation of the profile. Not only is the discrete approximation explicit, but calculus of variations of the previous section revealed that the RSVS has an alternative analytical and explicit solution to the RSVS problem. This dual implicit-explicit property makes the RSVS process flexible: the implicit equation is extremely general; while explicit representations permit conversions, reprocessing and translation.

3.6.3.1 The RSVS as a link between density and level set methods

Like density based STO methods, the RSVS uses a physically meaningful design variable in the form of the volume fraction, conceptually very similar to density for a non-porous domain. The implicit definition of the RSVS profile is also reminiscent of level set methods which define a crisp profile as the zero level of an level-set function (LSF), with various methods available for evolving the geometry. This link to both of the main methods of STO makes the RSVS a good candidate to interpret density based results into level sets and vice-versa.

For a single RSVS cell the similarity to level-set methods can be established by considering a level-set method (LSM) with a conical LSF. Solutions to the analytical RSVS process will also be level sets of the conical LSF. Changing the design variables of the RSVS corresponds to choosing a level set of an underlying implicit function. It remains unclear what the shape of a general LSF matching the RSVS governing equation would be.

An alternate interpretation lies in the form of solution conditions for optimisation problems: the Karush-Kuhn-Tucker condition (KKT) conditions. The first order necessary condition states that the gradient of the Lagrangian of the optimisation problem must be 0. This condition can obviously be understood as finding the 0 level set of a LSF defined by the first derivative of the Lagrangian; however from a practical standpoint it is unclear what the implications of this form may be.

The challenge of VOS based smooth parameterisation is intrinsically that of establishing a link between density and level set methods of geometry representation. In that goal the RSVS is very similar to the volumetric parameterisation of Hall et al. [155]. In that parameterisation method a smooth volume function was explicitly patched between cells; from this level set function a level set which matches the volume fractions would

be identified using a triangle or square marching procedure. The RSVS adopts instead a formulation with intrinsic smoothness properties, and resolves the implicit system to form a smooth geometry without explicitly defining a level set, even if there may be an underlying implicit level set formulation.

3.6.3.2 Alteration of properties for additional requirements

The RSVS generates profiles made of circular arcs with first derivative continuity (G_1). While this is sufficient for optimisation purposes, design applications usually expect at least curvature continuity (G_2) for smooth shapes. To alter it without loosing the compactness of the parameterisation the energy functional being minimised by the profile can be modified. An alternative to length minimisation could be the thin plate energy function presented in Equation 3.52; if used as the objective of the governing equation it would lead to profiles with higher order derivative continuity.

$$J[f] = \int_{y_0}^{y_1} \int_{x_0}^{x_1} f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2 dxdy \tag{3.52}$$

While it would lead to different profiles, other elements of the parameterisation process would remain the same: this could lead to the definition of a family of parameterisation method based upon parametric active contours and VOS design variables. The formulation used in this work has the benefit of allowing the construction of sharp corners with few design variables. This means that an optimiser can naturally construct sharp corners an ability that would be lost by increasing the smoothness of the profiles generated by the RSVS.

CHAPTER 4

PROPERTIES OF THE TWO DIMENSIONAL RSVS

So far the restricted snakes volume of solid (RSVS) has not been used in an optimisation framework and only its ability to match an interpolated profile has been tested. For the RSVS to be suitable for aerodynamic optimisation, it must have a smooth response of the generated profile with changes in the aerodynamic design variables. Without this smooth relationship, gradient based optimisers will not be capable of exploring the design space defined by the RSVS, making it unsuitable for aerodynamic optimisation. Beyond the smoothness of the response, work by Masters et al. [9] highlighted the impact of this "basis" of the parameterisation on the effectiveness and efficiency of the optimisation process. The basis for a small variation of the RSVS, shown below in Figure 4.1, is seen to be oscillatory, an undesirable property.

The goal of this chapter is two-fold: first is a study of the oscillatory response through analytical and numerical analyses. Then this chapter explores ways to tailor the response to make the RSVS parameterisation effective in a wide range of optimisation problems through smoothing and a hierarchical formulation of the design variables.

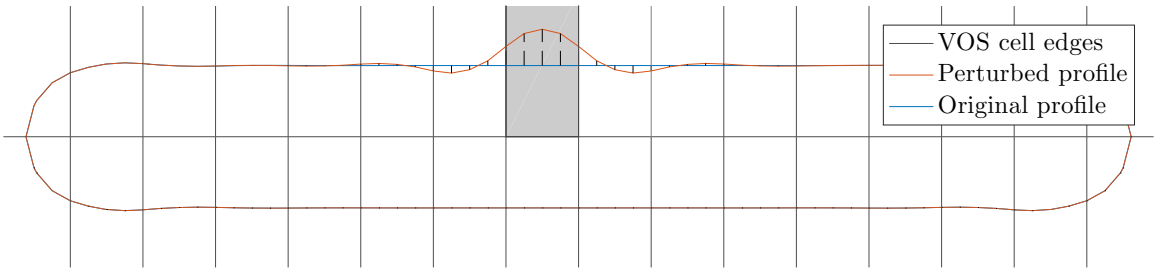


Figure 4.1: Oscillatory change in RSVS profile with a change in a VOS value.

4.1 Response to Changes in Area Constraint Using Calculus of Variations

Performance of aerodynamic optimisation frameworks is highly dependent on the behaviour and flexibility of the parameterisation method. in particular its geometric response to small changes of the design variables. The analytical formulation of the length minimisation enables the calculation of this shape response due to small changes in the volume constraints. This section shows that for well parameterised profiles the response is a quadratic spline.

4.1.1 Calculation of Profile Response to a Change in Volume

For any parameterisation the response of the geometry to small parameter changes is of interest. In order to analyse the analytical response of the RSVS to a change in volume fraction a variable substitution must be specified. This variable substitution is shown in Figure 4.2 and in Equations 4.1 to 4.6. Instead of measuring the response to small changes of the area A, it is measured against c_y, the y-position of the centre of the arc of circle defined by a single cell RSVS condition. This is to circumvent the implicit formulation of A with regard to the integration parameters.

Equations 4.1 and 4.2 define pinned boundary conditions that are used in this analysis. Equation 4.3 and 4.4 describe the relationship between the area A and the parameter used in its place for the analysis of the mode shapes, c_y. Figure 4.2 displays those variables graphically and shows the consistency between A and c_y response shapes.

The following additional assumptions are made to simplify the derivation:

- d is assumed not to change;
- c_x is assumed to be 0 for this analysis.

These correspond to limiting movements to be vertical.

In order to get the exact response of the analytical profile to a change in the constraint it would be necessary to calculate the explicit relationships of c_x, c_y and d with respect to the constraint value A. Unfortunately Equation 4.4 is not invertible, however, for shape optimisation, the normalised shape of the response is of more interest than the amplitude. It is important to note that A the constraint value changes monotonously with c_y and does not depend on x; this means the shape of the response is completely captured by the term c_y. The bounds and limits of the monotonic map f between c_y and A are presented in Equation 4.7.

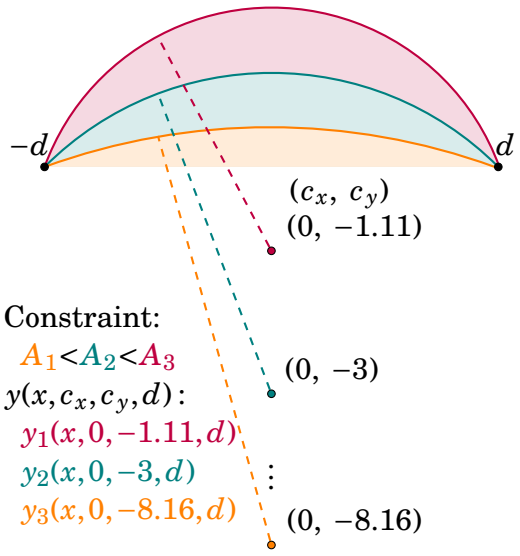


Figure 4.2: Arc of circle used in the derivation of the response to changes of volume fraction.

$$\alpha = d \text{ and } y(d) = 0 \quad \Rightarrow \quad \left(-c_y\right)^2 + (d - c_x)^2 = \frac{1}{\lambda^2} \tag{4.1}$$

$$\beta = -d \text{ and } y(-d) = 0 \quad \Rightarrow \quad \left(-c_y\right)^2 + (d + c_x)^2 = \frac{1}{\lambda^2} \tag{4.2}$$

$$\int_{-d}^{d} \left(c_y \mp \frac{1}{\lambda} \sqrt{1 - \lambda^2 (x - c_x)^2} \right) dx = A \tag{4.3}$$

$$A \equiv f\left(c_y\right) = (d^2 + c_y^2)\tan^{-1}(d/c_y) - c_y d \tag{4.4}$$

$$\left(y - c_y\right)^2 + (x - c_x)^2 = d^2 + c_y^2 \tag{4.5}$$

$$y\left(x, c_x, c_y, d\right) = c_y - \sqrt{c_y^2 + d^2 - (x - c_x)^2} \qquad x \in [c_x - d,\ c_x + d] \tag{4.6}$$

$$A = f\left(c_y\right) \quad \text{where} \quad f : [0, \infty) \rightarrow \left[\frac{\pi d^2}{2},\ 0 \right) \tag{4.7}$$

The shape of the response can be extracted by taking the derivative of y (Equation 4.6) with respect to parameter c_y.

$$\frac{\partial y}{\partial c_y} = 1 - \frac{c_y}{\sqrt{c_y^2 + d^2 - (x - c_x)^2}} \tag{4.8}$$

To get the normalised shape response the expression for $\partial y / \partial c_y$ must be divided by its maximum value, which occurs for $x = c_x$.

$$\phi\left(x, c_x, c_y, d\right) = \frac{\frac{\partial y}{\partial c_y}}{\max\left(\frac{\partial y}{\partial c_y}\right)} = \frac{\sqrt{c_y^2 + d^2 - (x - c_x)^2} - c_y}{\sqrt{c_y^2 + d^2 - (x - c_x)^2}} \frac{\sqrt{c_y^2 + d^2}}{\sqrt{c_y^2 + d^2} - c_y} \tag{4.9}$$

4.1.2 Shape of the Response for a Straight Edge

Unfortunately the shape function ϕ described by Equation 4.9 cannot be easily interpreted, it is necessary to explore its behaviour for the limits of c_y. The limit of ϕ as c_y tends to 0 is straightforward and well behaved (Equation 4.10), however the limit as c_y

tends to infinity is indeterminate. To improve the clarity of this process, it is useful to perform the substitutions outlined in Equation 4.11.

$$\lim_{c_y \to 0} \phi\left(x, d, c_x, c_y\right) = 1 \tag{4.10}$$

$$\begin{aligned} g_1\left(x, c_x, c_y, d\right) &= \sqrt{c_y^2 + d^2 - (x - c_x)^2}\sqrt{c_y^2 + d^2} \\ g_2\left(x, c_x, c_y, d\right) &= \sqrt{c_y^2 + d^2} \\ g_3\left(x, c_x, c_y, d\right) &= \sqrt{c_y^2 + d^2 - (x - c_x)^2} \end{aligned} \tag{4.11}$$

Using these substitutions Equation 4.9 can be conveniently rearranged in the following form, which avoid much of the indetermination when taking the limit of this equation as c_y tends to infinity.

$$\phi\left(x, c_x, c_y, d\right) = \frac{g_1 - c_y g_2}{g_1 - c_y g_3} = \frac{g_1^2 - \left(c_y g_2\right)^2}{g_1^2 - \left(c_y g_3\right)^2}\frac{g_1 + c_y g_3}{g_1 + c_y g_2} \tag{4.12}$$

$$\lim_{c_y \to \infty} g_1 = \lim_{c_y \to \infty} c_y g_2 = \lim_{c_y \to \infty} c_y g_3 = \lim_{c_y \to \infty} c_y^2 \to \lim_{c_y \to \infty} \frac{g_1 + c_y g_3}{g_1 + c_y g_2} = 1 \tag{4.13}$$

$$\lim_{c_y \to \infty} \phi = \lim_{c_y \to \infty} \frac{g_1^2 - \left(c_y g_2\right)^2}{g_1^2 - \left(c_y g_3\right)^2} = \lim_{c_y \to \infty} \frac{c_y^2 d^2 - c_y^2 (x - c_x)^2 - d^2 (x - c_x)^2 + d^2}{c_y^2 d^2 - d^2 (x - c_x)^2 + d^2} \tag{4.14}$$

$$\lim_{c_y \to \infty} \phi = 1 - \frac{(x - c_x)^2}{d^2} \tag{4.15}$$

The final result (Equation 4.15) shows that the surface response when the profile approaches a straight line tends to a parabola; this is represented in figure 4.3. This result proves analytically the smooth response of the RSVS to small changes of volume of solid (VOS). This result is important as a well parameterised shape has low levels of curvature in each cell, indeed the refinement methods developed in Section 4.4 aim to reduce the curvature of the profile in each cell.

4.1.3 Normal Modal Response to a Change in Volume Fraction

The analysis of the surface response to small design variable changes has been carried out in the direction of the y-axis which is aligned with the volume constraint. For the purpose of shape parameterisation it can be more beneficial to consider movement normal in the local axis system of the profile. This is done by projecting the gradient with regard to c_y onto the normal of the profile. This is done by taking the dot product between the

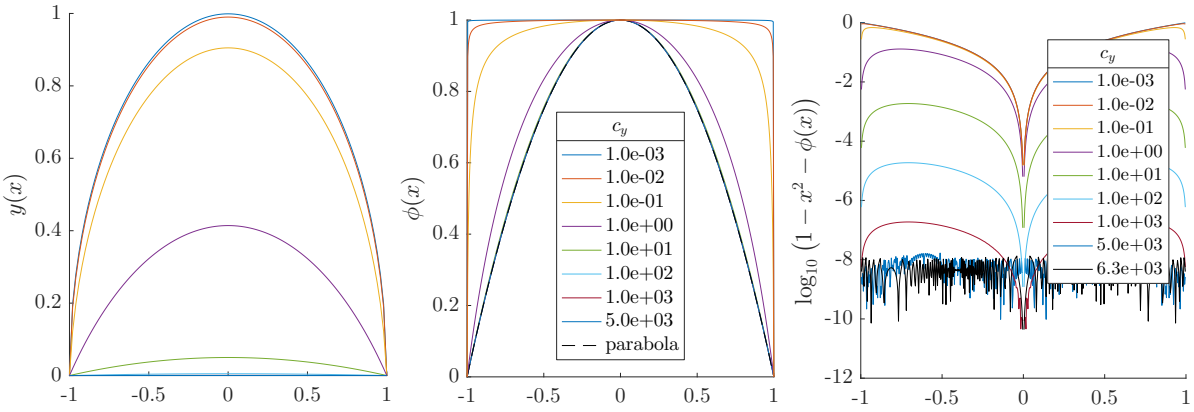


Figure 4.3: Modal response of the RSVS analytical response ($\phi(x, c_y)$) for different values of c_y (a surrogate for the area constraint value). This response is compared to a parabola for validation of the calculus of variations.

unit normal to the profile $\hat{\mathbf{n}}$ and the gradient in c_y. ϕ_N, the normal response of the profile, is rearranged into the canonical form of an ellipse in Equation 4.18. Figure 4.4 displays the shape of the normal response for different values of c_y.

$$\phi_N\left(x, c_x, c_y, d\right) = \left(0,\ \frac{\partial y}{\partial c_y}\right) \cdot \hat{\mathbf{n}} = \frac{\partial y}{\partial c_y} \Bigg/ \sqrt{\left(\frac{\partial y}{\partial x}\right)^2 + 1} \quad \text{with}: \mathbf{n} = \left(-\frac{\partial y}{\partial x},\ 1\right) \tag{4.16}$$

$$\phi_N\left(x, c_x, c_y, d\right) = \frac{\sqrt{c_y^2 - (x - c_x)^2 + d^2} - c_y}{\sqrt{c_y^2 + d^2}} \tag{4.17}$$

$$\left(\frac{1}{\sqrt{d^2 + c_y^2}} x - \frac{c_x}{\sqrt{d^2 + c_y^2}}\right)^2 + \left(\phi_N + \frac{c_y}{\sqrt{d^2 + c_y^2}}\right)^2 = 1 \tag{4.18}$$

This section has developed the analytical response of curves of minimum length under area constraints for small changes of the area constraints. These properties are necessary to ensure efficient integration of the RSVS with gradient based aerodynamic optimisation frameworks. It has been shown that for a profile with low curvature in the parameter cells, the response will tend to a continuous quadratic spline. Practical uses of RSVS result in profiles with low curvature in each design cell which means that the response of the RSVS to a small disturbance will be close to a quadratic spline, a well understood class of functions. Section 4.2 validates these results on profiles generated by the RSVS process.

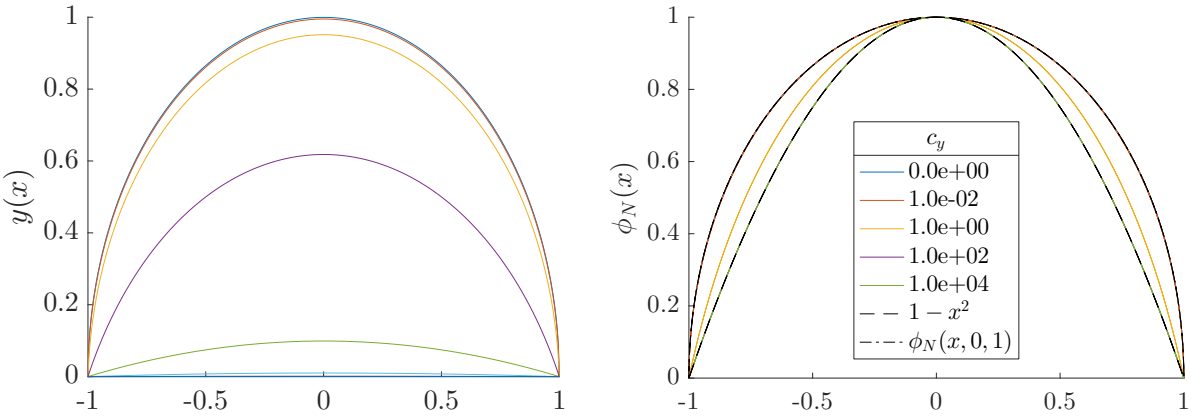


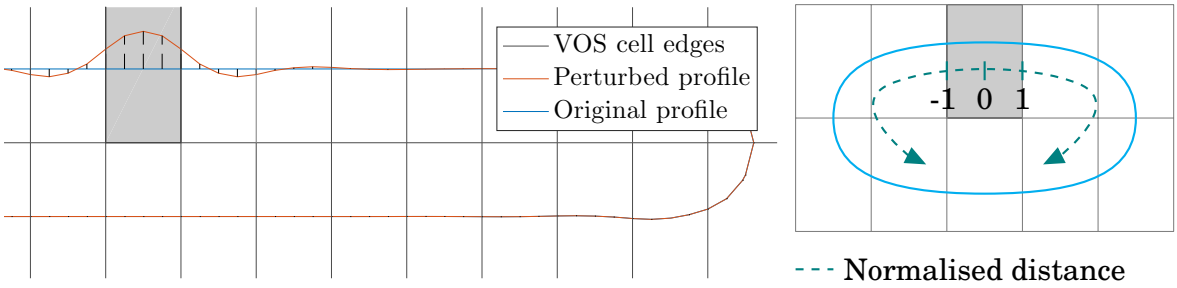
Figure 4.4: Plots of the analytical normal response ϕ_N (right) for various analytical single cell solutions $y(x)$ (left).

4.1.4 Limitations of Analysis Using Calculus of Variations

In order to compute the response of RSVS profiles built with more than one constraint the analytical formulas generated above need to be assembled into splines; the evolution of the conditions between patches must be explored. Section 3.2.2.1 formulated the boundary conditions needed to find an analytical solution to the RSVS governing equation; however these equations are not readily manipulated. Like the generation of RSVS profiles, analysis of the response of the boundary points is more practical through discrete solutions to the RSVS formulation achieved using restricted snakes. The next sections will study the response of discrete profiles defined by groups of VOS variables and establish a model based on the calculus of variations to smooth and tailor the basis function of the RSVS.

4.2 Oscillatory Design Variable Response Analysis and Smoothing

Behaviour of parameterisation methods is governed by the shape of the basis function: the change in geometry due to a change in control parameter. A number of studies into aerodynamic optimisation have compared the performance of different parameterisation methods and highlighted some of the desirable characteristics [9, 57, 164]. Desirable characteristics include: smoothness, appropriate scaling and compactness. This section aims to extract and modify the basis functions of the RSVS parameterisation in order to improve the behaviour of gradient based optimisation relying on this parameterisation.



(a) Oscillatory response (b) Normalised distance process

Figure 4.5: Oscillatory response of the RSVS and the representation of the normalisation process used in the subsequent analysis.

While the previous section showed that the response of the parameterisation was expected to be parabolic it could not conclude anything about the combined response of a profile crossing multiple VOS cells. By plotting the change in profile for a small change in volume fraction, Figure 4.5 reveals that the RSVS has an oscillatory response. Oscillatory bases have been shown to be very detrimental to optimiser behaviour [9, 57] and must be smoothed out for a gradient based optimiser to efficiently explore the design space. Before a smoothing process can be derived, the shape and intensity of those oscillations have to be quantified. Sections 4.2.1 and 4.2.2 present the derivation of an analytical expression for the oscillatory basis function.

Once the oscillations have been quantified smoothing of the perturbation had to be carried out; Smoothing of the response is done by smearing volume fraction steps to neighbouring cells along the profile. Simpler approaches to smoothing such as a simple isotropic smearing of volume fraction across nearest neighbours is not practical with the RSVS approach and does not lead to smooth perturbations. The analytical expression for the basis developed in Section 4.2.2 is used to compute how much 'smearing' is required to remove the oscillations. In subSection 4.2.4 the smoothing process is shown to improve both the convergence behaviour and the final optima on aerodynamic test cases.

4.2.1 Sensitivity of the Profile to the Volume of Solid

The parametrised contour is the result of an optimisation method where the volume fraction is a constraint on the design (see Equation 3.15). This formulation means the change in position of the profile due to a change in the volume fraction can be calculated analytically through a local sensitivity analysis. This approach means that the calculation of derivatives benefits from a wealth of previous research into sensitivity

analyses for sequential quadratic programming (SQP) algorithms [89, 90].

In order to exploit adjoint methods for flow solvers, the matrix governing the response of the profile to changes of design variables needs to be available. This implies that the quantity of interest is the Jacobian of optimum snaxel positions, $\mathbf{P}^*$, with regard to the requested volume fractions, $\mathbf{a}$. This requirement reduces to obtaining the Jacobian of the optimum distances along the edges $\mathbf{d}^*$ as shown by Equation 4.19.

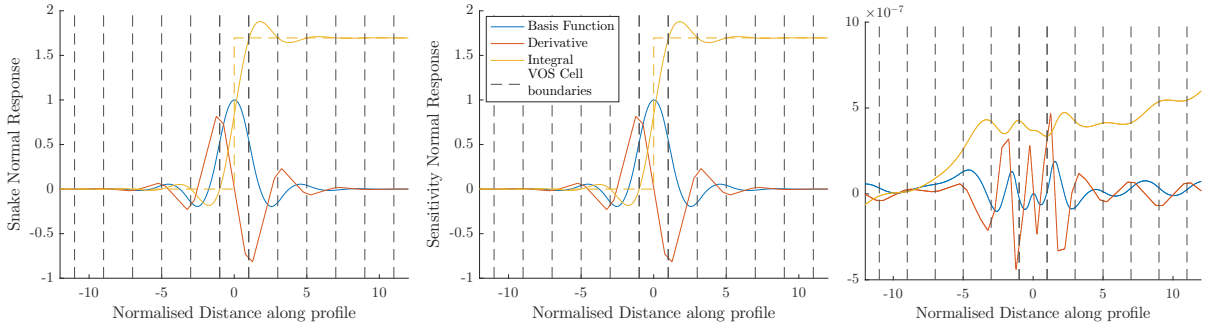
$$\nabla_{\mathbf{a}} \mathbf{p}_i^* = \nabla_{\mathbf{a}} \left(d_i^*(\mathbf{a}) \Delta \mathbf{g}_i + \mathbf{g}_{i,1} \right) = \Delta \mathbf{g}_i \nabla_{\mathbf{a}} d_i^*(\mathbf{a}) \tag{4.19}$$

This observation is important as the distances are the decision variables of the SQP algorithm used in the parametrisation process and the volume fractions the constraints. This means that the derivatives of interest are the sensitivities of the decision variables to the constraints for an optimal solution. As a consequence results from Buskens and Maurer [90] for the sensitivity analysis of Non-Linear Programs can be applied to the current problem. Sensitivity analysis equations from Buskens and Maurer [90] yield Equation 4.20. While the derivation of this formula is not presented in this work, the important observation is that all right hand side terms can be evaluated analytically for the final snaking profile. This gives the analytical value of $\nabla_{\mathbf{a}} \mathbf{d}$ which is used to get the change in position of the snaxels.

$$\begin{pmatrix} \nabla_{\mathbf{a}} \mathbf{d} \\ \nabla_{\mathbf{a}} \boldsymbol{\lambda} \end{pmatrix} = - \begin{pmatrix} \mathbf{H}_{\mathbf{d}} \mathscr{L} & \nabla_{\mathbf{d}} \mathbf{h}^T \\ \nabla_{\mathbf{d}} \mathbf{h} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \left(\nabla_{\mathbf{d}}^T \nabla_{\mathbf{a}} \right) \mathscr{L} \\ \nabla_{\mathbf{a}} \mathbf{h} \end{pmatrix} \tag{4.20}$$

Equation 4.20 is essential to the integration of an adjoint solver with the RSVS parameterisation. In effect matrix $\nabla_{\mathbf{a}} \mathbf{d}$ establishes a linear relationship between the VOS cell and the snaxel positions, from there it is straightforward to establish the relationship to the mesh points. The resulting matrices project mesh point adjoint sensitivities onto the RSVS basis.

The basis calculated using the analytical sensitivity in Equation 4.20 is compared to a basis function generated by perturbation of the design variable in Figure 4.6. In this figure the response is shown in a 1-dimensional space normal to the profile. This 1-dimensional space is normalised such that the centre of the disturbance is at the origin and that VOS cells are of length 2. The positional and volume errors are displayed in Figure 4.6c, and are of the order of 10^{-7} of the original response. This is comparable to the order of convergence of the position 10^{-8} divided by the order of the VOS perturbation



(a) Basis Function using Snakes (b) Basis Function using Sensitivity (c) Error between modes

Figure 4.6: Comparison of the basis functions extracted from the snaking process and the sensitivity analysis

10^{-1}. The equivalence between sensitivity and perturbation modes validates the use of Equation 4.20 to compute small changes in the restricted snake (r-snake).

Observing Figures 4.6a and 4.6b, the gradient of the basis is linear in each VOS cell with matching end points. This indicates that the basis functions resulting from the snaking process are quadratic splines. This matches the results of Section 4.1 which showed that for portions of profile with little curvature the VOS response is parabolic. This property is observed independently of the sampling rate of the curve.

This section has used sensitivity analysis of the SQP to compute and visualise the response of RSVS profiles to small changes in volume fraction. Combined with the observations of Section 4.1, it can be concluded that the basis of the RSVS can be modelled, to a high degree of precision, as an oscillatory parabolic spline. These oscillations are detrimental to performance in design optimisation frameworks and must be smoothed out. In order to remove those oscillations the next section will develop a method for computing the values of the parameters which define this parabolic spline.

4.2.2 Analytical Parabolic Basis Function Calculation

This section will derive the exact basis function of the RSVS for the special case of an infinite RSVS profile with equally spaced area constraints. The disturbance of a single central VOS cell in the centre of an infinite line is considered for the derivation of the response. The previous sections have outlined how the basis function is close to a piecewise continuous quadratic spline for arbitrary profiles with low curvature in each cell; this insight allows the derivation an analytical equation.

The disturbed VOS cell is centred on 0 and spans from $-d$ to d; all cells are of width

$2d$. The quadratic spline ($\phi_A(x)$) is composed of one polynomial patch ($p_j(x)$) in each VOS cell. The general formulation of the spline is presented in Equation 4.21; its polynomial pieces are detailed in Equations 4.22, 4.23 and 4.24.

$$\phi_A(x) = p_j(x - d(2j-1)) \quad x \in [d(2j-1),\ d(2j+1)) \quad \text{and} \quad \forall j \in \mathbb{Z} \tag{4.21}$$

$$p_j(x) = a_j x^2 + b_j x + c_j = \left[x^2,\ x,\ 1\right]\left[a_j,\ b_j,\ c_j\right]^T \tag{4.22}$$

$$P_j(x) = \frac{1}{3}a_j x^3 + \frac{1}{2}b_j x^2 + c_j x \qquad x \in [0, 2d) \tag{4.23}$$

$$\gamma_j = \left[a_j,\ b_j,\ c_j\right]^T \tag{4.24}$$

Each polynomial patch, $p_j(x)$, corresponds to the j^{th} VOS cell following the profile and $P_j(x)$ is its integral. The j index starts from 0 at the central VOS cell. d is half the length of a VOS cell, chosen to be 1 for the calculations in all subsequent figures. ΔV is the change in volume requested in the central cell. Some of the algebra in the following sections is best approached in matrix form, for this purpose let γ_j be the vector of coefficients corresponding to polynomial $p_j(x)$.

There are three sequential conditions which exist for all parabolic patches except the central one. Two arise from the continuity of the spline: derivative and polynomial patches value must be equal at the edge of the VOS cells. The parabolic patches also need to meet the VOS requirement ensuring that a change in area only happens in the central cell. These three conditions are the same as those presented for the many-constraint calculus of variations boundaries specified in Section 3.2.1; however because the spline is now parabolic derivation of the coefficients can be performed. These three conditions are expressed for the j^{th} polynomial in Equation 4.25, it is useful to express those in matrix form. These repeated conditions are expressed as a sequence defining the polynomial coefficients γ_j in terms of γ_{j-1}. These three conditions apply to all the polynomial pieces after the 0^{th}. Because there are three conditions and three coefficients per patch these are enough to define uniquely the parabolic arc provided that coefficients for the previous patch are known.

Repeated conditions for polynomial patches $\forall j \in \mathbb{N}^*$:

$$\left\{\begin{array}{rl} \text{Volume constraint}: & \left[P_j(x)\right]_0^{2d} = 0 \\ \text{Derivative continuity}: & \dfrac{\partial p_j}{\partial x}(0) = \dfrac{\partial p_{j-1}}{\partial x}(2d) \\ \text{Continuity}: & p_j(0) = p_{j-1}(2d) \end{array}\right\} \rightarrow \gamma_j = [\mathbf{C}]\gamma_{j-1} = \begin{bmatrix} -6 & \dfrac{-9}{4d} & \dfrac{-3}{4d^2} \\ 4d & 1 & 0 \\ 4d^2 & 2d & 1 \end{bmatrix} \gamma_{j-1} \tag{4.25}$$

These three conditions are not sufficient to calculate all the coefficients defining the polynomials: additional conditions are required to define the coefficients of the central polynomial patch (the 0^{th}). The first additional requirement is for the gradient of the disturbed cell to be 0 at the origin; this is a consequence of the symmetry of the basis and is observed empirically in Figure 4.6. Secondly the polynomial patches are required to decay to 0 as the distance away from the origin tends to infinity. These rules are formalised into three types of properties: two initial conditions (relating to the volume disturbance and derivative condition in the central cell); three sequential conditions (relating to the C_1 continuity of the spline and the volume constraints of the RSVS) and finally a limit condition that the basis must tend to 0. All three types are necessary to find the right polynomial coefficients.

Initial conditions relate to the known properties of the central polynomial: the cell where the volume fraction 'disturbance' occurs. There are two initial conditions: the area and therefore the integral under the polynomial patch must match the size of the area disturbance (Equation 4.27) and the derivative is 0 at the origin (Equation 4.26). These two conditions combined with the sequential conditions presented above, leave a single unknown coefficient: a_0.

$$\text{Initial derivative condition}: \quad \frac{\partial p_0}{\partial x}(d) = 0 \quad \rightarrow \quad 2a_0 d + b_0 = 0 \quad \rightarrow \quad b_0 = -2da_0 \tag{4.26}$$

$$\text{Initial volume condition}: \quad \begin{array}{lll} \left[P_0(x)\right]_0^{2d} = \Delta V & \rightarrow & \dfrac{1}{3}a_0(2d)^3 + \dfrac{1}{2}b_0(2d)^2 + c_0(2d) = \Delta V \\ & \rightarrow & c_0 = \dfrac{\Delta V}{2d} + \dfrac{2}{3}d^2 a_0 \end{array} \tag{4.27}$$

To solve for the last unknown it is necessary to observe the behaviour of the spline as it tends to infinity; the basis decays to 0 as it tends to infinity. Equation 4.28 presents this condition formally. The matrix formulation of Equation 4.24 is important as it enables the use of eigenanalysis to solve for the limit behaviour of the polynomial patches. The limit condition is required to close the system as all the coefficients except a_0 have been accounted for. The end condition is: as j tends to infinity the value of the polynomial p_j tends to 0 where $x = 2d$.

Limit condition :

$$\begin{aligned} &\lim_{j\to\infty} p_j(2d) = 0 \to \lim_{j\to\infty} \left[4d^2,\ 2d,\ 1\right][\mathbf{C}]^j \gamma_0 = 0 \\ &\qquad \to \left[4d^2,\ 2d,\ 1\right][\mathbf{V}] \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix} [\mathbf{V}]^{-1} \gamma_0 = [l_1,\ l_2,\ l_3]\gamma_0 = 0 \end{aligned} \tag{4.28}$$

$$\text{Solving for } a_0 : \qquad a_0 = \frac{-\Delta V\, l_3}{2d\left(l_1 - 2d\, l_2 + \frac{2}{3}d^2 l_3\right)} \tag{4.29}$$

[**V**] is the right eigenvector of matrix [**C**], it is used to find the limit of the $[\mathbf{C}]^\infty$ operation through eigenanalysis. Values l_1, l_2 and l_3 are defined by the value of d; these are used to compute a_0 using Equation 4.29. Coefficients beyond the first cell are derived using the sequence condition in Equation 4.33. Equations 4.30, 4.31, 4.33 and 4.32 summarise the calculation of the coefficients of the polynomial basis functions. These equations only hold for the positive side of the basis function; however it is straightforward to compute the negative side by symmetry.

$$\text{Initial derivative condition} : \qquad b_0 = -2da_0 \tag{4.30}$$

$$\text{Initial volume condition} : \qquad c_0 = \frac{\Delta V}{2d} + \frac{2}{3}d^2 a_0 \tag{4.31}$$

$$\text{Solving for } a_0 : \qquad a_0 = \frac{-\Delta V\, l_3}{2d\left(l_1 - 2d\, l_2 + \frac{2}{3}d^2 l_3\right)} \tag{4.32}$$

$$\text{Repeated conditions}: \quad \gamma_j = [\mathbf{C}]\gamma_{j-1} = \begin{bmatrix} -6 & \frac{-9}{4d} & \frac{-3}{4d^2} \\ 4d & 1 & 0 \\ 4d^2 & 2d & 1 \end{bmatrix} \gamma_{j-1} \tag{4.33}$$

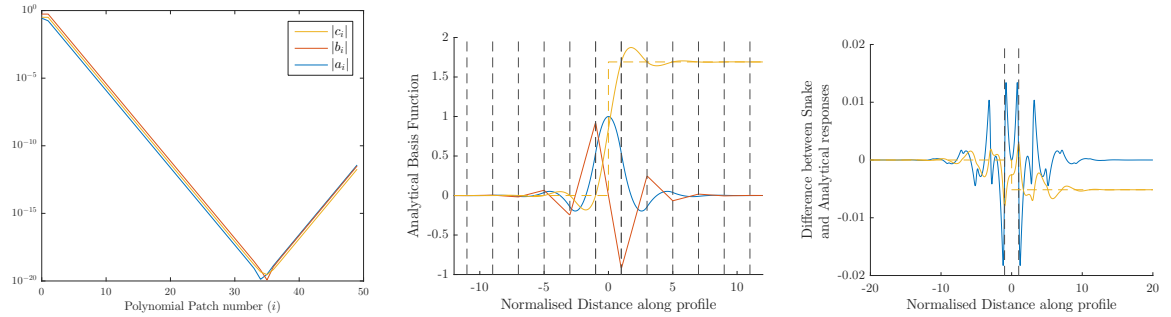
$$\text{Limit condition}: \quad \begin{cases} \lim_{j\to\infty} p_j(2d) = 0 \rightarrow \lim_{j\to\infty} \left[4d^2,\ 2d,\ 1\right][\mathbf{C}]^j \gamma_0 = 0 \\ \left[4d^2,\ 2d,\ 1\right][\mathbf{V}] \begin{bmatrix} 1 & & \\ & 0 & \\ & & 0 \end{bmatrix} [\mathbf{V}]^{-1} \gamma_0 = [l_1,\ l_2,\ l_3]\gamma_0 = 0 \end{cases} \tag{4.34}$$

The value of the polynomial coefficients are plotted in Figure 4.7a, the coefficients decrease exponentially, which translates into exponential decay of the basis function oscillations. It was found in the course of the analysis that the system which describes the basis function is limit stable. While coefficients exist that lead to an infinitely decaying basis function, in any evaluation of the numerical errors will eventually lead to diverging solutions. This is exactly the effect observed in Figure 4.7a, once the coefficients fall below 10^{-18} the precision of eigenanalysis and matrix inversion (used to define $[\mathbf{V}]$) become insufficient and the numerical error takes over leading to divergence of the basis function.

The analytical basis function is plotted in Figure 4.7b, while it looks very similar to the bases plotted in Figures 4.6a and 4.6b, Figure 4.7c shows that differences of the order of 1% exist between the theoretical basis and the computed response. The discrete nature of the snake process also affects the difference between theoretical and actual responses; looking at the gradients in Figure 4.6a it is clear that between the VOS cells their is another polynomial patch. This effect is the result of the discreteness of the r-snake process. These differences also have a knock on effect on the area constraint as these modes are normalised by their peak height rather than the area under the curve. Differences are also due to the assumption that the basis function stretches to infinity for the calculation of polynomial coefficients, practical profiles terminate leading to slightly different oscillations and coefficients.

4.2.3 Smoothing of the Basis Function

The analysis of the previous section revealed that the basis function generated by the RSVS parameterisation process is oscillatory. Previous comparative studies of aerodynamic shape parameterisation have shown that this is undesirable when using gradient



(a) Polynomial patch coefficients.

(b) Basis Function using Analytical calculations.

(c) Difference between analytical and experimental modes

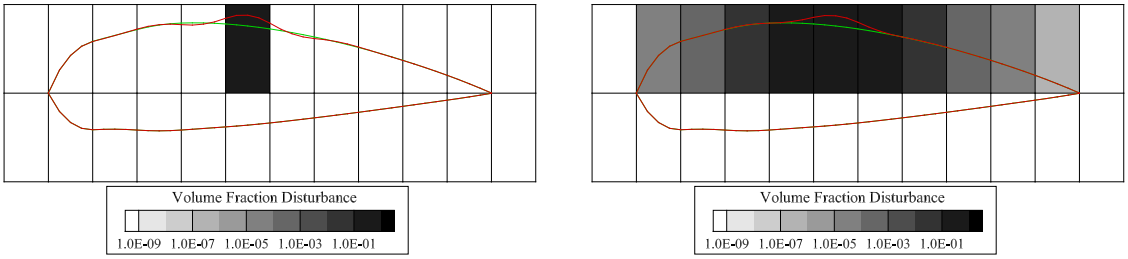
Figure 4.7: Analytical basis function.

based optimisation methods. The shape of the basis function has a large impact on convergence speed and quality of the optimum solution [9, 57], there is a need for a smooth and non-oscillatory basis functions. To improve convergence of design optimisation frameworks Jameson et al. have used a Sobolev inner product to smooth the gradient resulting from the adjoint [236]. Similarly a smoothing scheme is proposed for the RSVS which combines the responses of neighbouring volume fractions to smooth out oscillations.

Figure 4.8a shows a simple VOS disturbance that produces an oscillatory response. This response can be smoothed by mixing in some of the neighbouring volume fractions to counteract the oscillations, the single disturbance to the RSVS parameterisation becomes a smeared disturbance as shown in Figure 4.8b. Importantly the smoothing is done in surface-space rather than all the grid neighbours, this is because the oscillations follow the profile rather than cell connectivity. There are two aspects to the smoothing: that of the analytical response and that of the practical response of RSVS profiles; these are treated respectively in Sections 4.2.3.1 and 4.2.3.2.

4.2.3.1 Smoothing of the analytical response

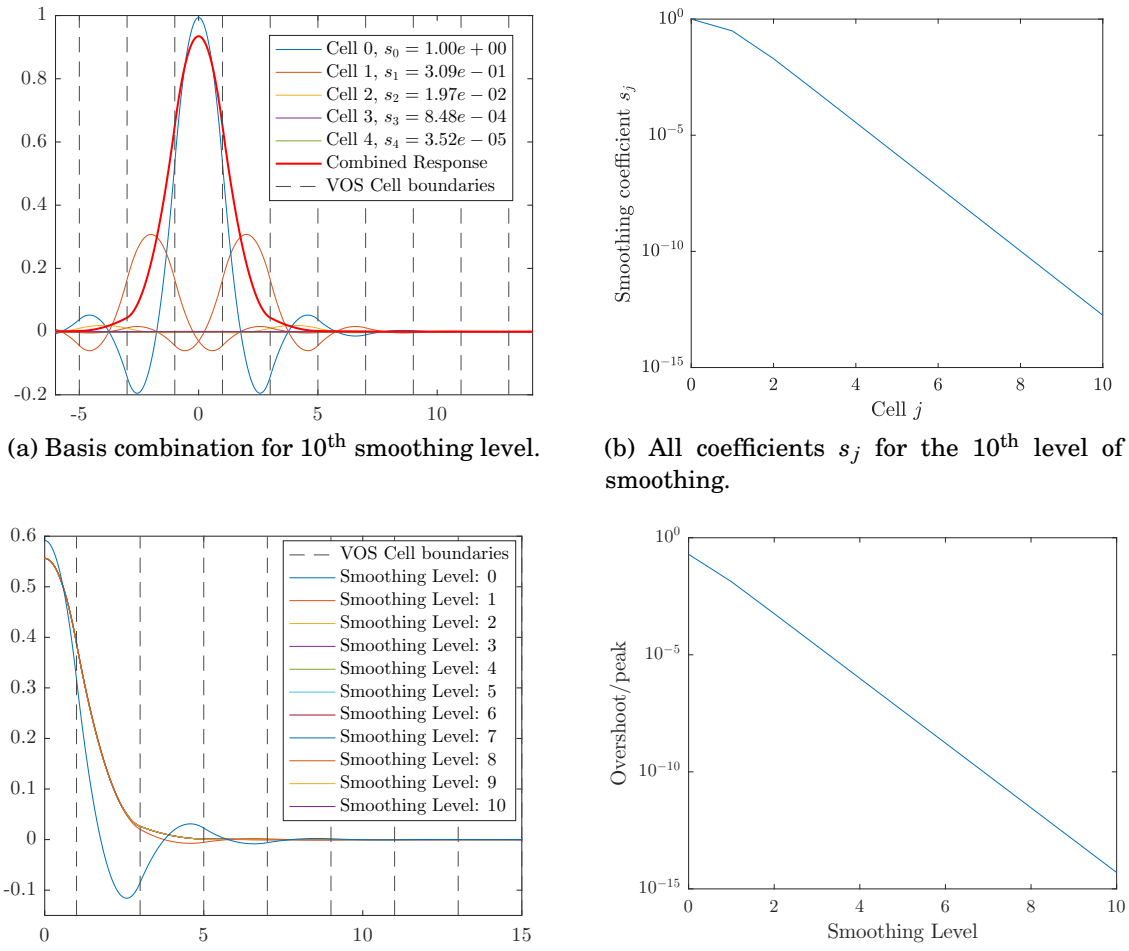
Approximate iterative smoothing To ensure a smooth response, neighbouring design variables need to be mixed together such that oscillations are exactly smoothed out; these analytical "mixing coefficients" are denoted by the symbol s_j. The exact height ($h_{p,j}$) and location ($x_{p,j}$) of the oscillation peak for each polynomial is calculated using the polynomial coefficients (γ_j) of the polynomial patches derived in the previous subsection. These two properties of the parabolic patch are used to calculate a volume fraction response (s_j) from a neighbouring cell which will smooth out the oscillation in that



(a) Single volume fraction perturbation.

(b) Perturbation of Volume fraction smoothed in surface-space.

Figure 4.8: Volume fraction perturbations for standard and smoothed gradient calculations.



(a) Basis combination for 10^{th} smoothing level.

(b) All coefficients s_j for the 10^{th} level of smoothing.

(c) Smoothed responses to unit volume disturbance.

(d) Maximum overshoot with smoothing level.

Figure 4.9: Smoothed analytical basis function

neighbour. This process can be repeated progressively, removing the oscillation in the following neighbour.

A given smoothing coefficient is derived by first calculating the current response in the first remaining oscillation in the j^{th} VOS cell. To ensure a smooth function, the influence of all previous mixed in responses must be taken into account when calculating s_j; this is done by using Equation 4.35 which sums all the polynomial coefficient multiplied by the appropriate previous mixing ratios. This process leads to a new, combined, polynomial coefficients for the oscillatory patch, which is smoothed by mixing in the j^{th} VOS cell following s_j calculated by Equation 4.38. The process of summing the responses is shown in Figure 4.9a. This progressively removes the overshoot of the basis function with an overshoot tending to 0 rapidly as seen in Figures 4.9c and 4.9d.

$$\text{using}: \quad s_0 = 1 \qquad \text{for all } j > 0 \quad s_j \text{ is derived using the following}:$$

$$\gamma_{s,j} = \begin{bmatrix} a_{s,j} \\ b_{s,j} \\ c_{s,j} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & \cdots & a_j & a_{j+1} & \cdots & a_{2j-1} \\ b_1 & b_2 & \cdots & b_j & b_{j+1} & \cdots & b_{2j-1} \\ c_1 & c_2 & \cdots & c_j & c_{j+1} & \cdots & c_{2j-1} \end{bmatrix} \left[s_{j-1},\, s_{j-2},\, \cdots,\, s_0, s_1,\, \cdots,\, s_{j-1} \right]^T \tag{4.35}$$

$$x_{p,j} = \frac{-b_{s,j}}{2a_{s,j}} \tag{4.36}$$

$$h_{p,j} = a_{s,j} x_{p,j}^2 + b_{s,j} x_{p,j} + c_{s,j} \tag{4.37}$$

$$s_j = \frac{h_{p,j}}{h_{p,0}} \frac{\left(d + |d - x_{p,j}|\right)}{d} \tag{4.38}$$

Exact "one-shot" smoothing Looking at Figure 4.7a it is clear that all the three polynomial coefficients forming γ_j decay geometrically and uniformly away from the disturbed VOS cell. Using this observation, it should be possible to find "mixing" coefficients similar to those specified above that yield a compactly supported basis function (a basis which decays to 0 in finite distance). This compactly supported basis exists as all the oscillations are self similar with decaying sizes: two or more bases can be combined to cancel out oscillations completely.

The geometric change in coefficient also implies that, except for the first coefficients (γ_0), the coefficient vector γ_j is an eigenvector of the sequential condition matrix ([**C**] Equation 4.25). The imposition of the limit condition in Equation 4.28 turns out to be equivalent to finding a value of a_0 such that [**C**]γ_0 is an eigenvector of [**C**] which

corresponds to a negative eigenvalue with an absolute value below 1. For a value of $d = 1$ the value of the eigenvalues (λ_C) and eigenvectors $[\mathbf{V}]_C$ of the sequential conditions are shown in Equation 4.39. Using the properties of eigenvectors the sequential condition can be expressed as a simple scalar product (Equation 4.40).

$$\lambda_C = \begin{bmatrix} -\sqrt{3}-2 \\ \sqrt{3}-2 \\ 0 \end{bmatrix} \qquad [\mathbf{V}]_C = \begin{bmatrix} 1 & 1 & 1 \\ \frac{1}{3}(2\sqrt{3}-6) & -\frac{1}{3}(2\sqrt{3}+6) & -4 \\ -\frac{1}{3}(2\sqrt{3}-2) & \frac{1}{3}(2\sqrt{3}+2) & 4 \end{bmatrix} \tag{4.39}$$

$$\gamma_{j+1} = \lambda_2 \gamma_j = \left(\sqrt{3}-2\right)\gamma_{j-1} \quad \forall j \in \mathbb{N}^* \tag{4.40}$$

This formulation of the sequential condition allow an exact smoothing coefficient (s_e) to be calculated. This coefficient is found by looking to set $\gamma_{s,2} = \mathbf{0}$; if the coefficient vector falls to 0 at any point along the spline, by virtue of the sequential condition, the remaining basis will be 0. Using the approach of summing polynomial coefficients to compute a smoother basis of Equation 4.35, the expression for this exactly smoothed polynomial patch is presented in Equation 4.41. The expression relies on an unknown exact smoothing coefficient applied in both neighbours of a central disturbance.

$$\begin{aligned} \gamma_{s,2} = s_e\gamma_1 + \gamma_2 + s_e\gamma_3 \quad &\rightarrow \quad \mathbf{0} = s_e\gamma_1 + \lambda_2\gamma_1 + s_e(\lambda_2)^2\gamma_1 \\ &\rightarrow \quad \mathbf{0} = \left(s_e + \lambda_2 + s_e(\lambda_2)^2\right)\gamma_1 \\ &\rightarrow \quad s_e = \frac{\lambda_2}{1+\lambda^2} \end{aligned} \tag{4.41}$$

$$s_e = \frac{\sqrt{3}-2}{1+\left(\sqrt{3}-2\right)^2} = \frac{1}{4} \tag{4.42}$$

The formula of Equation 4.42 allows an analytical expression for the exact smoothing coefficient to be derived; it is found that a VOS response of 25% in the first neighbours will smooth out the oscillations on Cartesian grids. On anisotropic grids additional work is required to obtain consistent mode shapes and mode size. To smooth out oscillations on anisotropic grids the smoothing values, either exact or iterative, must be scaled to reflect the relative sizes of the cells and their subsequent response due to a change in VOS. The next section develops a method to ensure that the relative responses are smooth and appropriately scaled for optimisation.

4.2.3.2 Smoothing of practical profiles

The current smoothing analysis has been limited by the consideration that the basis space is infinite in both directions and each cell is of equal span along the profile. To ensure effective smoothing for closed contours, fixed endpoints and variable sized cells, the coefficients need to be tailored to the profile and design variable layout. The basis function polynomial patches were calculated for 1-Dimensional intervals of equal lengths; while this provides sufficient accuracy to gain insight into the nature and intensity of the oscillations, it is not sufficient to effectively smooth out practical profiles.

This subsection considers the coefficients needed to obtain a smooth change in RSVS geometry as a result of a unit change in VOS in cell J. Cell J is taken to be in the centre of $2m+1$ cells that make up the profile, with j indexing those cells. To resolve the difference between analytical assumptions and practical profile, an additional coefficient ($q_{J,j}$) is calculated in Equation 4.43 to modify the ideal smoothing coefficients calculated earlier (Equation 4.38). This new coefficient introduces length and volume scaling into the smoothing formulation to take into account cell and profile geometry. In each VOS cell the coefficient becomes dependent on the length of the profile (l_j) and the volume of the cell (V_j) relative to the previous cell ($j-1$) along the profile.

$$q_{k,j} = \frac{l_j}{V_j} \frac{V_{j-1}}{l_{j-1} + \sqrt{c_{j-1} V_{j-1} \delta}} q_{k,j-1} \tag{4.43}$$

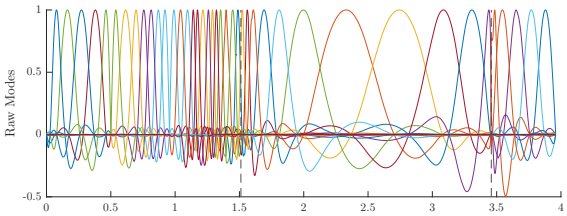
The final step of the smoothing process is to scale all modes so that they have either a consistent height response ($\rho_{k,h}$ Equation 4.44) or a consistent volume response ($\rho_{k,v}$ Equation 4.45). This leads to the final VOS step being defined by the values of ρ_j (Equation 4.46) for a smooth, grid adapted and consistently scaled set of basis functions around the profile.

$$\text{Height scaling}: \qquad \rho_{k,h} = \frac{V_0}{l_0} \tag{4.44}$$

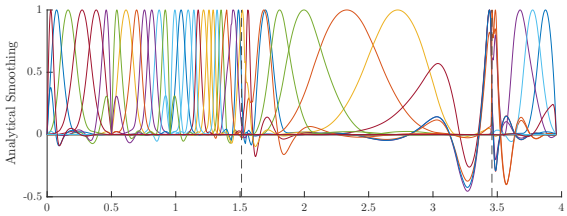
$$\text{Volume scaling}: \qquad \rho_{k,v} = \frac{1}{\sum_{i=1}^{m} V_i \, q_{j,i} \, s_i} \tag{4.45}$$

$$\rho_{k,j} = \rho_{k,(h|v)} \, q_{k,j} \, s_j \qquad \forall j \geq 0 \tag{4.46}$$

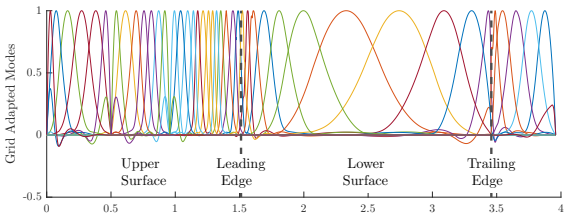
Using the process of mixing in neighbouring volume fractions outlined in Figures 4.8b and 4.9 with the coefficients calculated using Equations 4.38 and 4.43, the effective response of a profile designed on an anisotropic grid is shown in Figure 4.10. Figures 4.10a, 4.10b and 4.10c show the effect of changing the smoothing coefficients to alter



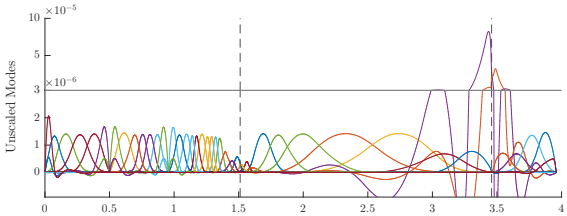
(a) Normalised raw response.



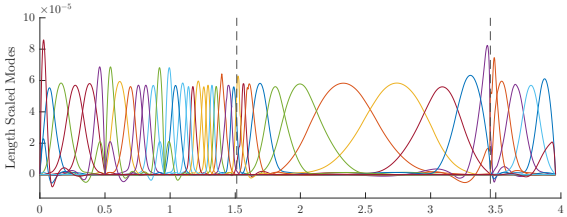
(b) Smoothed response with analytical coefficients (Equation 4.38).



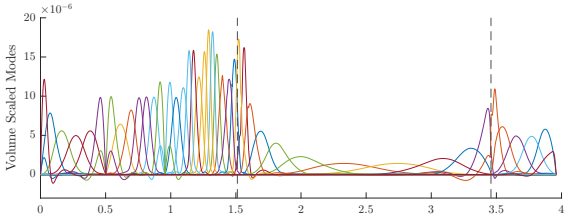
(c) Normalised smoothed response with grid adaption (Equation 4.43).



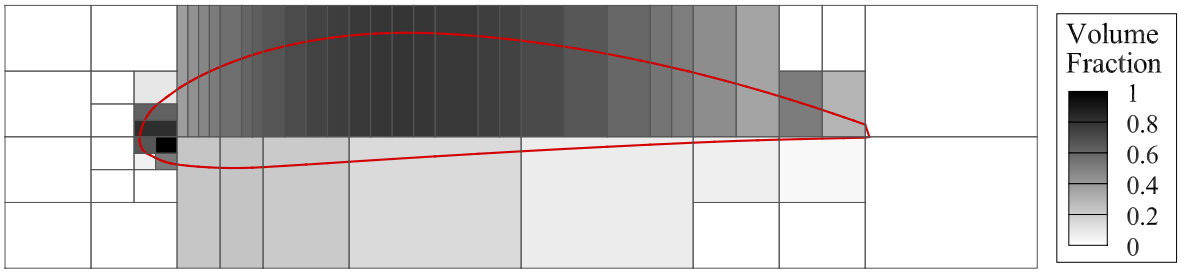
(d) Smoothed response with grid adaption (Equation 4.43).



(e) Length scaling of the smooth response.



(f) Volume scaling of the smooth response.



(g) Grid, volume fractions and profile from which the modes were evaluated.

Figure 4.10: Modal Responses to small changes in the volume fraction for an anisotropic VOS grid.

the shape of the mode. Table 4.1 shows the importance of both the analytical smoothing and the grid adaption as each reduces the average and extreme overshoot cases. Figure 4.10d shows that the modes resulting of the smoothing process can correspond to very different response sizes; Figures 4.10e and 4.10f show response height and response volume scaling respectively. The choice between these two scaling is dependent on the objective function: for the inverse design cases volume scaling was used as the objective

Table 4.1: Overshoot comparison for the modes in Figure 4.10

Overshoot Data	Mean	Median	Minimum	Max	Mean(Log10)
Raw response (Figure 4.10a)	2.345E-01	2.156E-01	1.354E-01	4.936E-01	-0.645
Analytical smoothing (Figure 4.10b)	7.827E-02	2.326E-02	2.174E-04	4.551E-01	-1.803
Grid adapted smoothing (Figure 4.10c)	1.793E-02	4.177E-03	2.226E-04	9.119E-02	-2.313
Linear programming (Figure 4.12)	4.425E-03	6.269E-11	4.304E-11	1.763E-01	-9.263

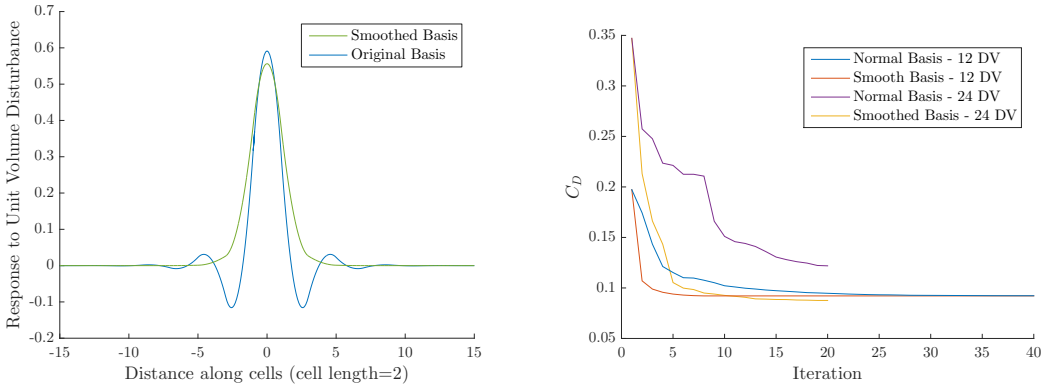
is dependent on volume changes; for aerodynamic optimisation length scaling was used as it is standard practice in the field.

While in theory this combination of smoothing coefficients should lead to an overshoot tending to 0 with increasing smoothing level as shown in Figure 4.9d, in practice the overshoot is reduced to the order of 10^{-2} of the response of the central basis. Numerical results for practical responses is shown in Table 4.1;

4.2.4 Impact of the Design Variable Smoothing on Aerodynamic Optimisation

The basis function smoothing discussed in this section was implemented in a gradient based optimisation framework. In this section only the impact of the design variable smoothing developed in the previous section is considered; a detailed description of the framework is provided in Chapter 5.

Figure 4.11 shows the effect of basis function smoothing on an aerodynamic test case at Mach 2 with 12 and 24 VOS cells. For the 12 design variables cases the smoothing allows an improvement in the final result of 1.5 drag counts, however there is a dramatic improvement in the convergence behaviour of the optimisation: the final result is reached after 8 iterations instead of 40. These effects are even more pronounced for larger number of design variables, with 24 design variables the smoothed basis allows convergence in 20 iterations, while the normal basis displays very poor convergence behaviour. These results show that without modification of the underlying design variables the convergence behaviour for a given optimiser can be improved significantly by combination of variables, while this was used here only to smooth design variables smarter adaptive combinations of design variables could be implemented.



(a) Normal and Smoothed Basis functions

(b) Aerodynamic Optimisation results for two basis functions

Figure 4.11: Mach 2, constant area, Aerodynamic Optimisation with and without basis function smoothing for an axis ratio of 1.5

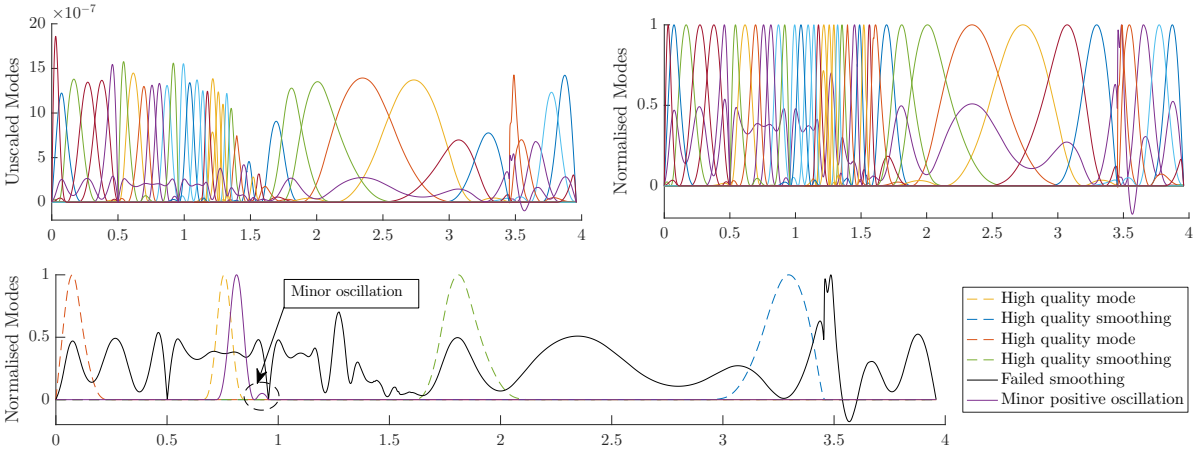


Figure 4.12: Modes smoothed using the linear programming approach.

4.2.5 Perfect smoothing of Coefficients on Individual Sensitivity Matrices

One alternate approach to smoothing through basis combination was considered. Instead of relying on time consuming analytical derivation of the VOS response, the smoothing can be derived directly on the sensitivity response of individual r-snake profiles. The approach of this section trades some of the robustness of the previous methods, for some additional flexibility; notably its potential as a method for smoothing the response of three dimensional profiles. Indeed it is unclear if similar analysis will yield as robust a smoothing process when the RSVS is extended to three dimensions.

This alternate approach relies on directly smoothing the sensitivity of the profile

(matrix $\nabla_{\mathbf{a}}\mathbf{d}$ from Equation 4.20), allowing the actual response of the snaxel positions to changes in VOS values to be taken into account. The most appropriate basis combination can be computed on the fly by formulating the derivation of the smoothing process as a linear programming problem. For this linear optimisation problem, the objective is to minimise the movement requested by the smoothing coefficients; while constraining the response of the snaxels to be out of the profile removing any oscillation. The decision variables of this linear program are the VOS responses in all cells $\mathbf{q}_j$; unlike the previous method each smoothing mode is specific to a given VOS cell, hence the use of a subscript on the smoothed mode $\mathbf{q}_j$. The j$^{\text{th}}$ component of mode $\mathbf{q}_j$ is constrained to be equal to 1 ensuring that the response to a unit change of VOS in cell j is smoothed. This approach is presented in Equation 4.47 below.

$$\begin{aligned} &\text{For each cell } j \text{ a smoothed mode } \mathbf{q}_j \text{ solves}: \quad \min_{\mathbf{q}_j} \ \mathbf{f}^T\mathbf{q}_j \\ &\text{subject to}: \quad -[\nabla_{\mathbf{a}}\mathbf{d}]\mathbf{q}_j \le \mathbf{0} \\ &\qquad\qquad 0 \le q_{j,k} \le 1 \quad \forall k \ne j \\ &\qquad\qquad q_{j,j} = 1 \\ &\text{with}: \quad \mathbf{f} = [f_0, f_1, \cdots, f_j, \cdots, f_m] \qquad f_j = \sum_{i=0}^{n} \left([\nabla_{\mathbf{a}}\mathbf{d}]_{i,j}\right)^2 \end{aligned} \tag{4.47}$$

Modes generated using this type of smoothing are shown in Figure 4.12; these highlight how this approach can yield very compact modes with no oscillations in other direction. This excellent performance is shown in Table 4.1 where the linear programming smoothing outperforms the other types by 6 orders of magnitude on average. However the lower image of Figure 4.12 highlights how this approach is also prone to spectacular failures: the failed smoothing is significantly worse than anything produced by the analytical smoothing. Unfortunately this type of failure in the smoothing process is capable of stalling an optimisation process causing it to finish early; for this reason this type of smoothing was not used in optimisation with the two dimensional RSVS.

It was deemed unnecessary to resolve these issues, in light of the other, very effective, smoothing processes which were developed in the previous sections. It is likely that either some fall back logic or some slight modifications of the linear program could resolve these issues of robustness. It is however useful as a possible alternative for the smoothing of the three dimensional extension of the RSVS and any parameterisation relying on the RSVS framework but using a different energy functional (see Section 3.2.5 and 3.6.3).

4.3 Geometrical Discontinuity Around Changes of Topology

Despite the developments presented in the previous sections the geometric variation of the RSVS is not smooth everywhere in the design space: when the topology of a RSVS solution changes it can lead to a sudden change in profile. Another cause for discontinuities in the profile response is caused by the activation or de-activation of a VOS cell, i.e. turning it on, even infinitesimally. These discontinuities manifest as large changes in the profile geometry in response to tiny variations of the values defining an RSVS profile.

This section highlights those discontinuities in the profile response and presents some potential avenues to smoothing the RSVS response through those discontinuities. The discussion of this section relies on figures, like Figure 4.13, which represent RSVS profiles for a given layout of VOS cells as a surface, with the vertical axis proportional to the requested volume fractions. This format is used in Figure 4.13 to present the RSVS response for a single square VOS cell. Two main types of discontinuities exist in the RSVS response to changes of the volume fractions: topology change and constraint activity; each of which presents a different challenge.

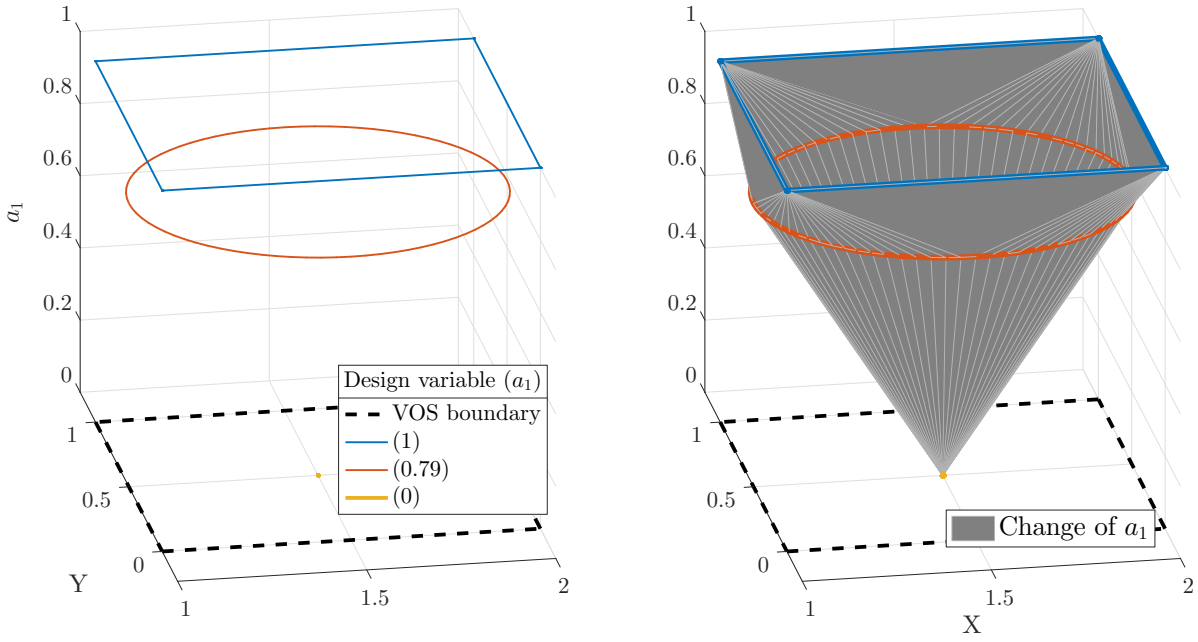


Figure 4.13: Surface generated by sweeping one VOS value through its range from 0 to 1; the “level set function” of RSVS is displayed.

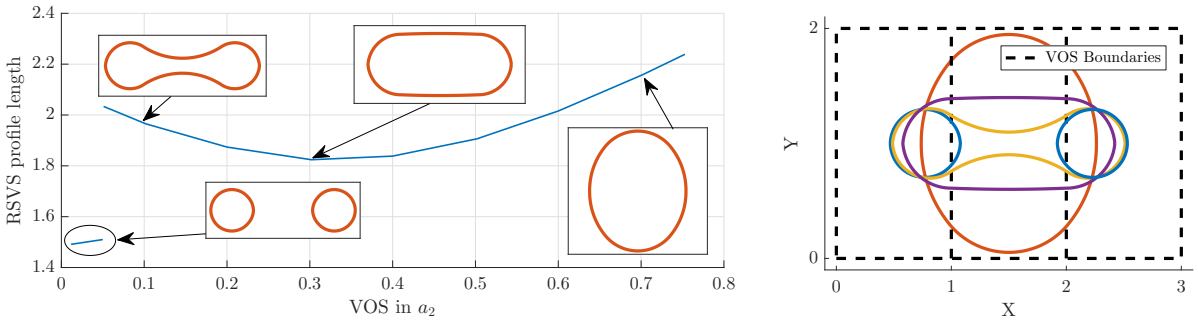


Figure 4.14: Length of the RSVS profile for a 3 by 1 VOS cell layout, showing a discontinuity in the length in the profile when the topology of the profile changes.

4.3.1 Discontinuity Through Topology Transition

The discontinuity resulting from a topology change is readily understood: as the connectivity of the profile changes, the positions which minimise the length of the profile change. In that respect, this discontinuity is the result of the choice of the RSVS to control topology implicitly combined with the use of length as the energy functional driving the shape of the profiles. This behaviour is the jump between two local minima, which are separated by their difference in profile connectivity. This jump between local minima is clearly seen in the sudden change in the length of the profile generated by the RSVS for a VOS in a_2 of 0.05 in Figure 4.14.

The geometric response surface of the 3 VOS cells is presented in Figure 4.15. The three cell system is seen to transition from a single body profile to a two body profile for an area constraint in the central cell (a_2) of 0.05. This discontinuity around the topology cut puts a strain on gradient based optimisers in an aerodynamic optimisation framework, as it is not visible in the sensitivity of the objective function. This usually means that the ability of gradient based optimisation to go through that topology change is not controlled, leading that to imperfect design space exploration.

This discontinuity highlights that, while the RSVS can generate very different profiles with very few design variables; the smoothness of the response decays as too many features try to be represented with few design variables. It would be desirable for a mechanism to exist which allows a smooth transition between the two local minima of the constrained length minimisation problem. Figure 4.16 shows a possible patch through the discontinuity of the RSVS by introducing an hypothetical design variable σ which controls the transition between the two profiles. In this context σ is defined on the interval $[0,\ 1]$ where it corresponds to the two body and single body profiles respectively.

While σ is not formally defined, the fact that a smooth transition can easily be

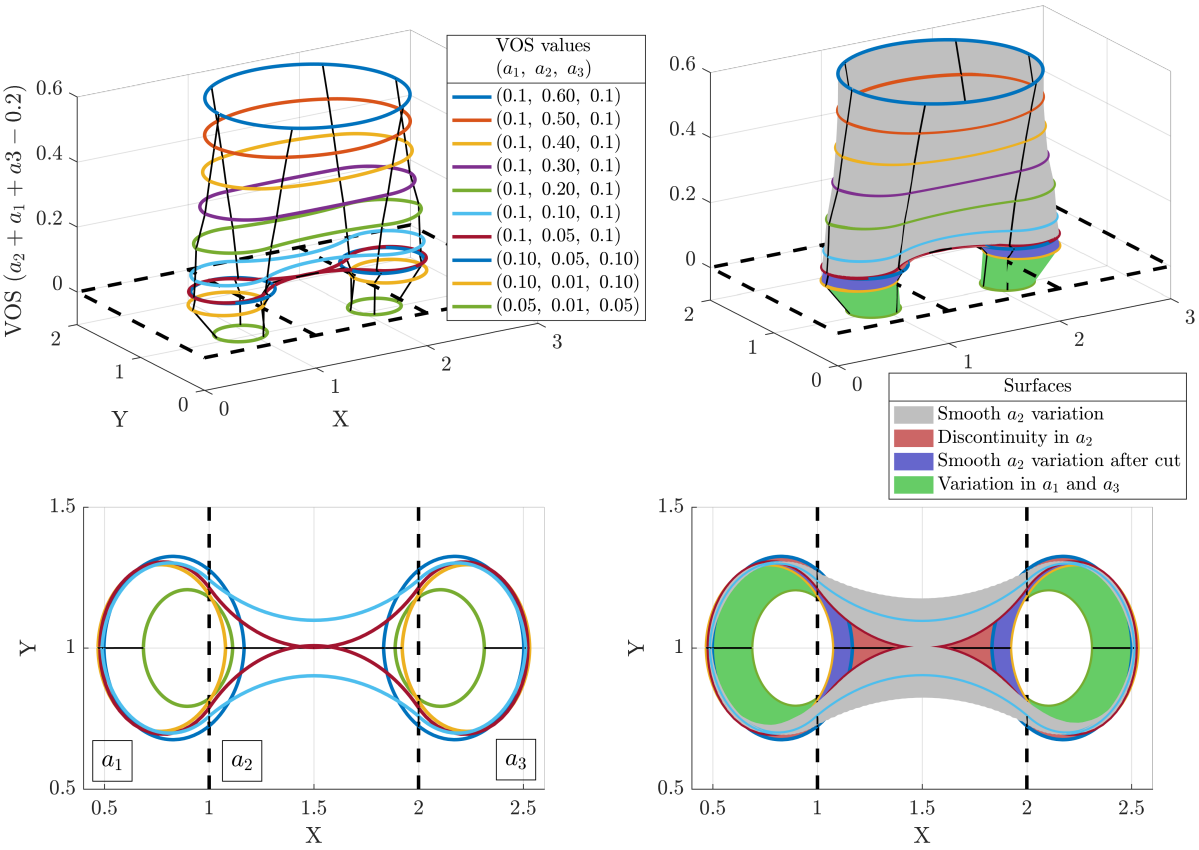


Figure 4.15: Discontinuity in profile shape around a topological change in profile generated with 3 VOS values.

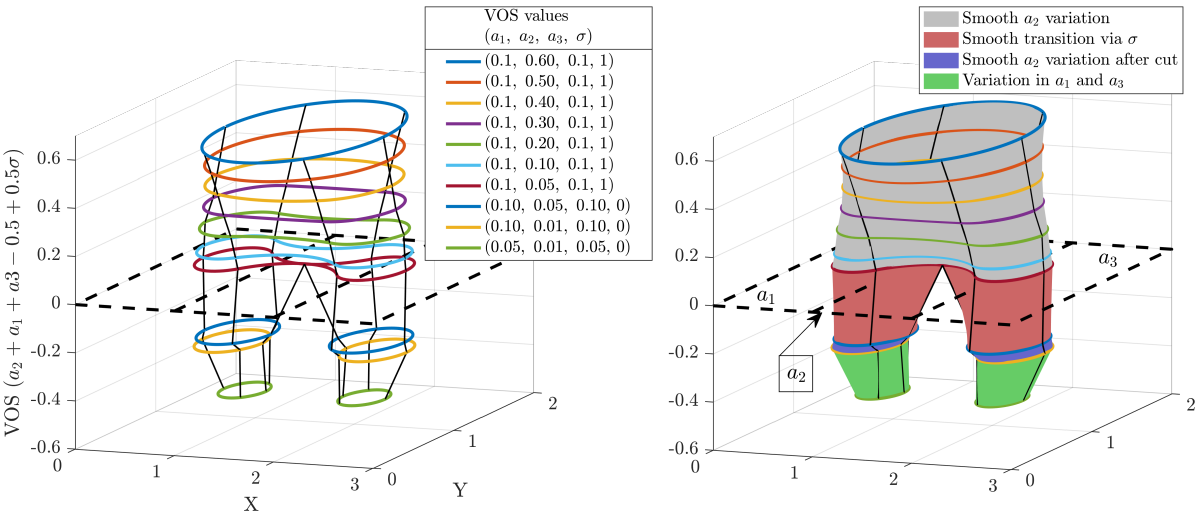


Figure 4.16: Discontinuity in profile shape patched by the addition of a σ design variable, providing a smooth transition between topologies, around a topological change in a profile generated with 3 VOS values.

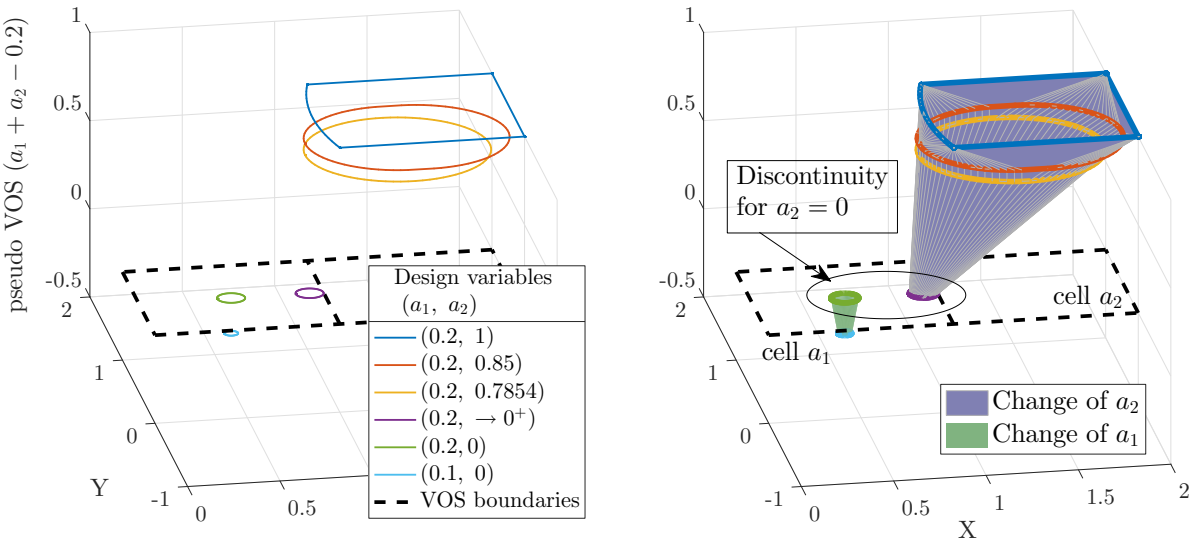


Figure 4.17: Analytical RSVS surface generated by sweeping two neighbouring VOS cells (a_1, a_2) through the range $(0.2, 1)$ to $(0.2, a_2 \rightarrow 0^+)$ and $(0.2, 0)$ to $(0.1, 0)$. This plot shows a discontinuity as a_2 reaches 0, manifesting as a sharp movement of the RSVS profile.

represented and visualised indicates that approaches based on the RSVS may permit a smoother transition between topologies. The discontinuity arises because the area constraints on a given layout do not provide sufficient control over a profile to differentiate between two profile topologies. This indicates that additional design variables are required either through a different type of control or constraint over the RSVS system or through additional VOS design variables. This observation can start guiding the design of RSVS layouts to ensure effective and sufficient parameterisation: separate objects should be represented by several cells each to ensure that they are not an artefact of the parameterisation.

4.3.2 Active Constraint Discontinuity

The second type of discontinuity in the RSVS response arises when VOS cells are activated or deactivated: when they go from 0 area constraint to a non-zero value. The response of the RSVS in this case is presented in Figure 4.17 for a 2 VOS cell system. It is clear that the response is smooth for the open interval $(0, 1]$ but shows a significant discontinuity when a_2 reaches 0: the circular profile generated by the RSVS jumps to the centre of cell a_1 when the value of a_2 reaches 0.

While the profile shape remains the same through the discontinuity the two profiles are distinct by their position and represent two solutions to the RSVS problem. Similarly

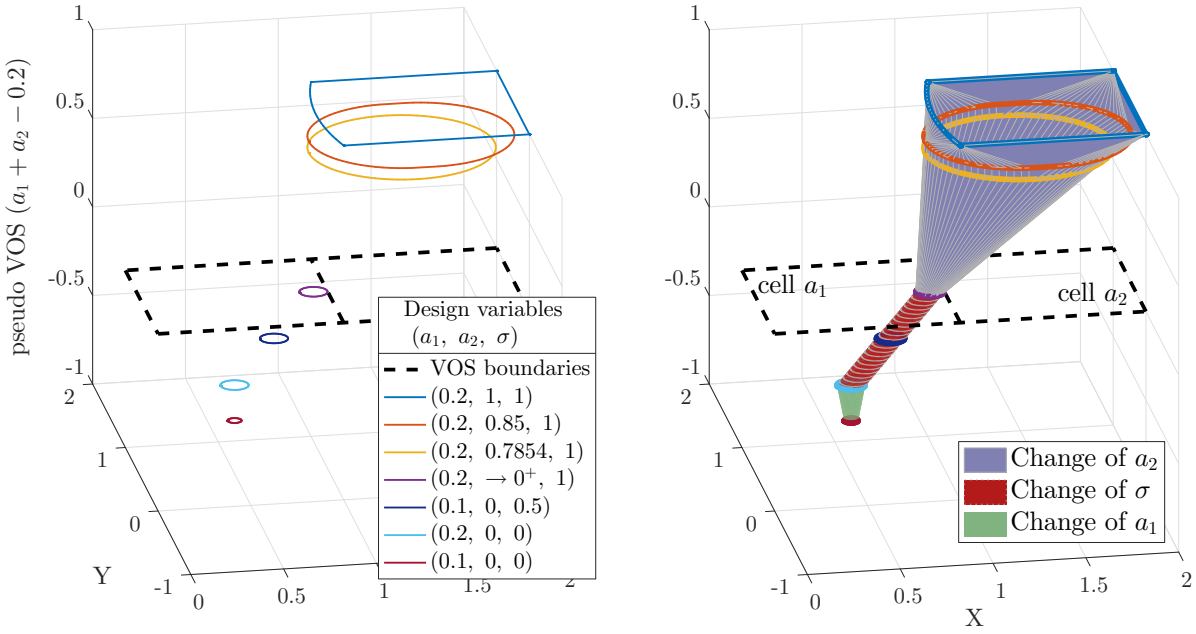


Figure 4.18: Potential approach to patch through the discontinuity by introducing an additional design variable σ which controls the distance to the edge between the two cells.

to the case of topological transition this jump is due to the area constraint providing insufficient information to differentiate the two cases. Indeed in the limit as a_2 tends to 0 a circular profile is fixed at the boundary between the cells; when a_2 reaches 0, the pinning effect of that very small VOS value is lost and the position of the profile within cell a_1 becomes uncontrolled. As was the case for the topological change, a smooth path through the discontinuity can be plotted (Figure 4.18) using an additional design variable σ.

4.3.3 Mitigation of the Discontinuities

Two discontinuities in the RSVS response have been identified when topology changes and when the activity of the area constraints changes. These discontinuities can be removed using an informal additional design variables σ; but the question of implementing this design variable remains. It is critical that this process does not restrict the flexibility of the RSVS or make its three dimensional extension impossible. This additional process should also leave most RSVS profiles unaffected only patching through the discontinuities.

Looking at the patches generated by sweeping σ in Figures 4.18 and 4.16; it appears

that the control that is not an area property but a property along a line. In the case of the topology cut, this property is whether a line linking two portions of the design space is inside or out of the profile. In the case of the de-activation of a RSVS cell the missing control relates to the position of the circular profile along that line.

A formal definition of σ could alter the RSVS formulation introducing additional energies to the objective function or constraints to allow control of these behaviours. Unfortunately this is likely to completely alter the philosophy of the RSVS and to require a very large number of additional controls to remove all the possible discontinuities. While two types of behaviours were presented, it is unclear how many exist and an enumeration of them and a proof that they are all mitigated is not feasible or necessary.

Instead of developing secondary control tools, it seems more scalable and robust to use the existing volume control to improve the traversal of these discontinuities. A smooth path through a given discontinuity can be defined by adding VOS design variables which are finer than the layout on which the discontinuity exists. With the appropriate layout of VOS a discontinuity can be mitigated during the optimisation process. Such a refinement of the VOS cells allows an optimisation process to progress smoothly through discontinuities which exist at coarser design levels.

Clearly such a refinement process also introduces its own discontinuities in the RSVS profile response; these additional discontinuities can be mitigated by additional refinement. While this recursive process seems endless, there will be a point for a practical optimisation process will not exploit these discontinuities any further at which point the existence of them does not matter and additional refinement is not required. In the next section, a refinement process is developed to patch through the discontinuity of RSVS response as it provided the clearest and surest way forward for the parameterisation. Such a refinement process fit neatly with the goals of keeping a very general parameterisation, and can exploit the great variety of layouts possible using the RSVS.

4.4 Design Variable Refinement

Refinement is proposed as a way to close the gap in performance between the RSVS and traditional aerodynamic parameterisation methods. Compared to these methods, one of the drawbacks of the RSVS parameterisation method is that a regular Cartesian VOS grid contains much less implicit information about aerodynamic problems. This means that the RSVS, while being more general than other parameterisations, also requires careful set-up of the design variable layout to tackle an optimisation problem

efficiently. This set-up might require the specification of regions with finer, coarser and distorted cells to achieve adequate performance. To alleviate this need for expert input, a hierarchical approach to the design variables is developed. These approaches start an optimisation problem with few design variables, this allows large but coarse changes to the design. As this process converges additional design variables are added allowing progressively finer and smaller scale changes to the design to be added.

Previous sections have developed and analysed the properties of RSVS geometries. The smoothness of the RSVS was established analytically regardless of the layout of area constraints. Changes in profile due to changes in VOS were shown to be continuous when the topology of the profile did not change and the active constraints remained the same. When discontinuities in the response appear, it is due to insufficient control through a given layout of VOS design variables. Refinement also offers a natural way to increase the control over the RSVS profile and provide a reduction of those discontinuities.

Hierarchical approaches by Anderson and Aftosmis [175] and Masters et al. [176] have accelerated and improved convergence on complex aerodynamic optimisation problems. Similar approaches have been successfully exploited in structural topology optimisation by Kim et al. to improve the performance of agent based optimisers [237] and by Bandara et al. to build a multi-resolution framework based on sequential shape and topology optimisation using subdivision curves [238].

The RSVS lends itself to such hierarchical approaches, the refinement of design VOS design variables is intuitive and exact. A locally adaptive hierarchical process is developed in sub-Section 4.4.2. Finally, the effects of local refinement are shown on inverse design and aerodynamic optimisation cases in Section 5.3.2 and 5.3 respectively.

4.4.1 Refinement of RSVS Design Grids

The process of implementing local refinement into the RSVS parameterisation relies on the accurate translation of design variables from coarse to fine fidelity levels. Exact translation of a profile from a coarse to a finer layout of design variables requires only that the coarse cell boundaries is present in the finer grid. The volume fractions on the finer grid can be calculated by overlaying the profile onto the fine grid and calculating the intersection of cells and profile. This approach is analytically exact: the underlying equivalent non-uniform rational B-Splines (NURBS) remaining the same; however in practice there are slight differences after refinement as the snaking grid is also refined, changing the discretisation of the profile.

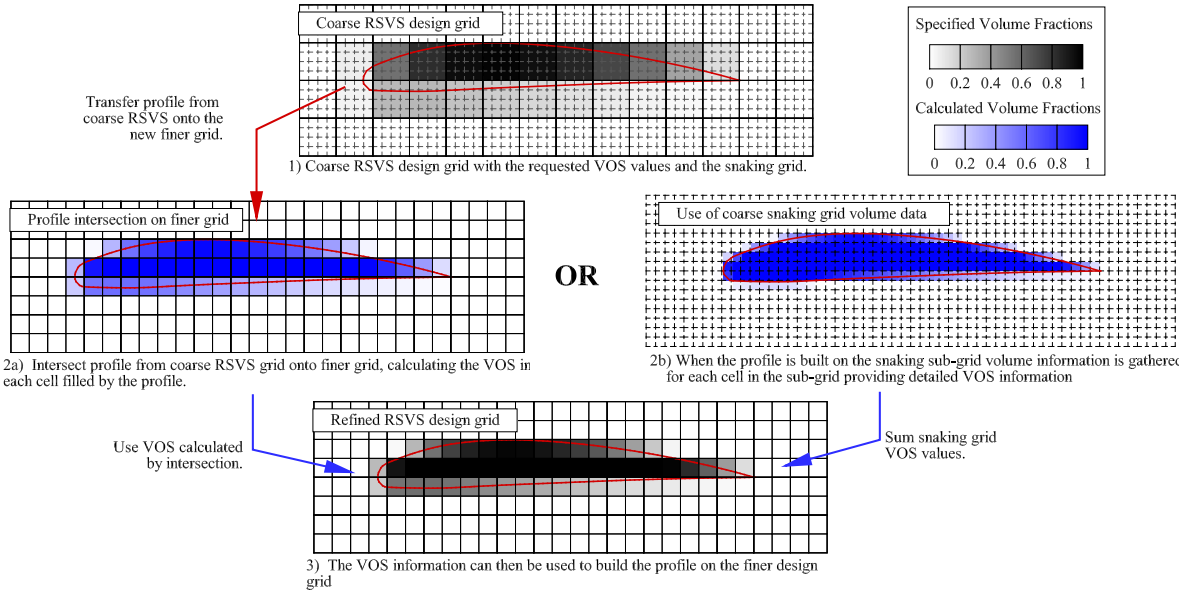
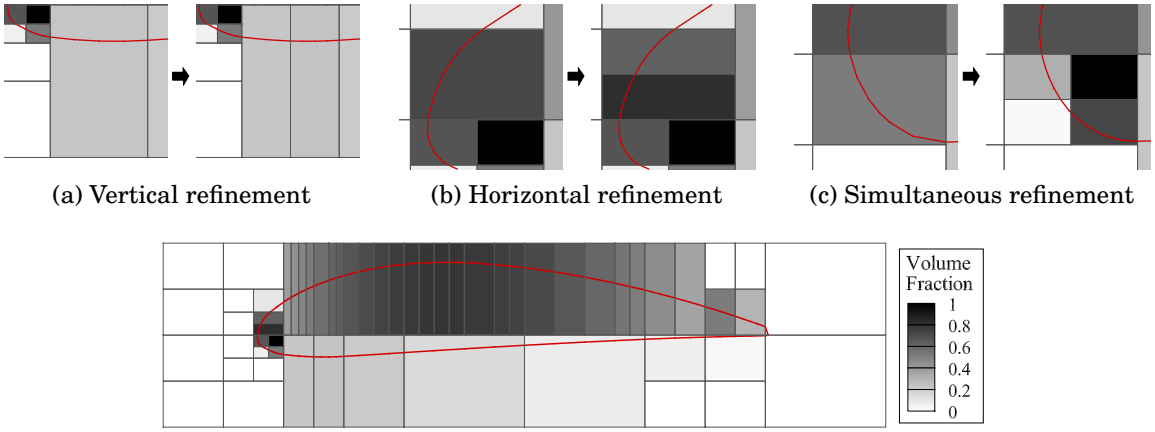


Figure 4.19: Process for the exact translation of a profile from a coarse RSVS design grid to a finer RSVS design grid using information from the snaking grid.

Calculation of containments on the new design grid can be skipped by using volume fractions defined on the snaking grid calculated during the evolution of an RSVS profile. Because in most cases the snaking grid is a 4 by 4 refinement of the design grid, exact profile translation can be done using information from the snaking grid for all VOS grid refinements up to 2 by 2. The different elements of this process are presented in Figure 4.19. To fully leverage the benefits of a hierarchical approach the algorithm needs to identify and refine regions of the design space where refinement will lead to an improved objective function.

4.4.2 Criterion for Refinement of Design Variables

Both uniform and local refinement of design variables have been considered by previous studies. Uniform approaches split all design variables regardless of their influence on the design; local methods aim to identify regions of the design which are more important to the reduction of the objective function and refine only those locations. Global refinement approaches have been very successful: the hierarchical approach based on subdivision curves presented by Masters et al. [176] show the best published results for the aerodynamics design optimisation discussion group (ADODG) NACA0012 Case 1 [179]. Results for local refinement have been mixed, with previous studies showing that improved performance could be achieved, but the increased complexity of the



(a) Vertical refinement (b) Horizontal refinement (c) Simultaneous refinement

(d) Grid, volume fractions and profile from which the splits were extracted.

Figure 4.20: Possible cell cuts under the local refinement algorithm.

optimisation method had a negative impact on the robustness of the process [176].

Previous parameterisations that were used with refinement were tailored specifically to the aerofoil optimisation problem; this meant that the need for local adaptation by the parameterisation was limited, and that global refinement was sufficient. The goal of the method presented here is to be able to tackle any shape optimisation problem with minimal tuning of the design variables by the user. For this general approach to be possible and efficient, the process developed must be able to adjust the local fidelity as the optimisation progresses.

This was achieved using a local refinement algorithm which selects which VOS cells should be refined and the direction in which they should be split. Eight different criteria were systematically tested on geometric and aerodynamic optimisation problems; they combine profile length and curvature with different types of normalisation. These different equations led to different patterns of refinement some with obviously degenerate behaviours, only the most efficient and robust of them is presented in the rest of this section. The most effective criterion refined cells with the most curvature of the profile. Curvature reliably indicates the difficulty the parameterisation is having to represent a given geometry because the minimum length objective tends to create the straightest line possible. Unlike previous methods [154, 176] which relied on adjoint sensitivities of the objective function, this criterion relies only on information provided by the shape parameterisation.

Equation 4.48 shows the calculation of the refinement criterion (τ_j) for cell j. Snaxel indices i_s and i_e refer to the first and last snaxel contained in the VOS cell. One critical

requirement for a refinement criterion is that its value must decrease as refinement is carried out, otherwise refinement can go into a self reinforcing loop which is unlikely to lead to desirable design variable layouts. The formula for coefficient τ_j is designed to balance the facts that:

- curvature tends to increase as the VOS cells get refined;
- length tends to decrease as the VOS cells get refined.

By scaling the curvature by the length, the potential for a feed back loop is negated and a balance between uniform refinement and refining regions of interest is struck.

$$\tau_j = \sum_{i=i_s}^{i_e-1} (\Delta s_{i+1}) \sum_{i=i_s}^{i_e} \left\| \frac{-\mathbf{p}_i(\Delta s_i + \Delta s_{i+1}) + \mathbf{p}_{i+1}\Delta s_i + \mathbf{p}_{i-1}\Delta s_{i+1}}{\Delta s_i^2 \Delta s_{i+1} + \Delta s_i \Delta s_{i+1}^2} \right\| \qquad (4.48)$$

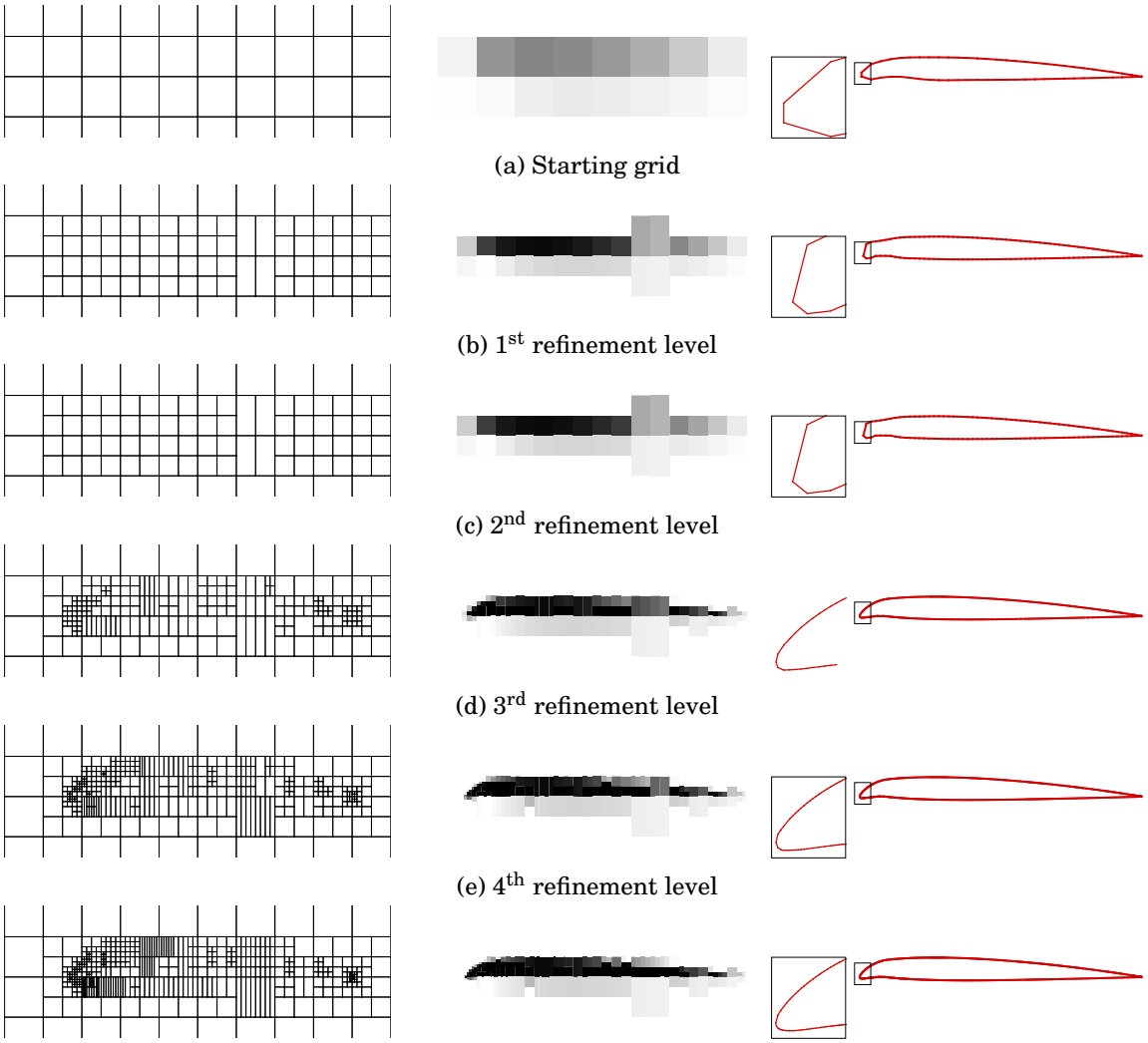
$$\text{with :} \qquad \Delta s_i = |\mathbf{p}_i - \mathbf{p}_{i-1}| \qquad \text{and} \qquad \Delta s_{i+1} = |\mathbf{p}_{i+1} - \mathbf{p}_i|$$

Once a cell has been marked for refinement, the orientation of the split must be decided. Selecting the refinement direction relies on a heuristic method which yields fine grids which still enable easy movement of the profile and high quality modes. Because of the cell-bounded volume of solid formulation transition of the profile from one cell to a previously empty cell can be discontinuous. In order to minimize such transitions new cell boundaries should be normal to the profile. To achieve this, cells are split in half in the same direction as each VOS cell edge which is crossed by the r-snake. This leads to three possible refinement outcomes which are presented in Figure 4.20.

4.4.3 Effect of Refinement on Benchmark Cases

In order to test the refinement process, the geometric inverse design performed in Section 3.3.3 to test the RSVS parameterisation was repeated. In this test, 65 NACA aerofoils are intersected with the RSVS grid, the profile is built and compared to the original aerofoil using the Kulfan's Wind Tunnel Tolerance (Kulfan's WTT). Instead of manually designing an appropriate grid, the process is started from a Cartesian, 2 by 9, VOS grid; and five refinement steps are performed. At each step all the profiles are intersected, the VOS calculated and the distance between target and generated profiles measured. The VOS grid is then refined according to the rules of Equation 4.48 and Figure 4.20 presented in the previous section. Numerical results of this process are shown in Table 4.2 and presented in Figure 4.21.

Table 4.2 shows an improvement of the geometric recovery of the profile with each refinement level. Finally at the 5^{th} refinement level; 95% of profiles are recovered to the required level, almost matching the result of the manually generated grid (96.7%).



(a) Starting grid

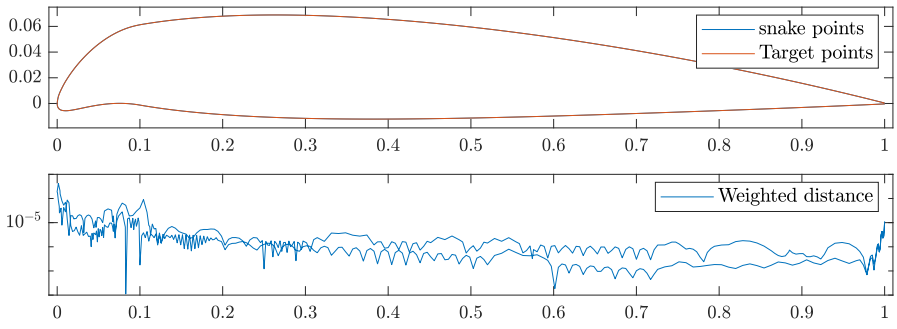
(b) 1st refinement level

(c) 2nd refinement level

(d) 3rd refinement level

(e) 4th refinement level

(f) 5th refinement level



(g) Profile and error at the final refinement level

Figure 4.21: Geometric inverse design of the NACA 3108 on anisotropically refined grids; from left to right: the VOS grid, the VOS values and the profile are shown at each refinement step.

Table 4.2: Result of the inverse design of 65 NACA airfoils with grids designed with automatic anisotropic refinement.

Refinement level	maximum of $\max(w_i n_i)$	Median of $\max(w_i n_i)$	Mean of $\max(w_i n_i)$	WTT satisfied	VOS variables	Mean Active VOS
0	1.93E-02	1.17E-02	1.17E-02	0%	18	18
1	2.15E-02	1.08E-02	1.19E-02	0%	68	34
2	6.28E-03	3.16E-03	3.08E-03	0%	142	68
3	3.57E-03	7.64E-04	8.72E-04	70.8%	215	97
4	1.48E-03	6.61E-04	6.90E-04	83.1%	346	142
5	1.41E-03	5.45E-04	5.67E-04	95.4%	434	179

The effectiveness of refinement on this case shows its ability to automatically adapt the parameterisation to a given problem, reducing the need for expert users to tune the parameterisation.

4.4.4 Limit of the Refinement Process

Considering the refinement as a tool for "patching" discontinuities in the RSVS response, it is clear that it is imperfect: there is no guarantee that a given discontinuity will become smooth through a single refinement step. However it is interesting to note that line control, which was presented as a straightforward way to understand the continuation through a given discontinuity, can be seen as control over an infinitely thin VOS cell. Such a 1 dimensional VOS cell can be the result of infinite uni-directional refinement, highlighting a potential theoretical link between the two ideas. While not explored further in this work, the idea of a limit of the refinement of design variables could allow better control of positional discontinuities in the RSVS response to changes of volume fraction.

In the context of a design optimisation process the effectiveness of refinement depends on its ability to improve the objective function with each refinement step. This behaviour is in fact due to refinement "unlocking" a new part of the design space leading to performance improvement. There is a limit to this process: as the true, physical optimum is reached and no further improvement will yield any improvements. Theoretically, when using gradient based optimisation, this can be detected through the evolution of the optimality of a design. If refinement causes a reduction in the optimality of a profile it is beneficial; otherwise, if the optimality remains through refinement, the process can be stopped as the refinement is converged. In that respect this allows to stop a refinement process confident that no further gains are likely.

4.5 Summary

For a design optimisation framework to successfully exploit a parameterisation method the geometric response to changes of control parameters needs to be "well behaved". The set of desirable properties in aerodynamic shape optimisation (ASO) has been the subject of a number of studies [8, 9, 57, 163, 164] which led to recommendations regarding the compactness, the periodicity and the scaling of design variables.

Careful analysis of the RSVS basis showed it to be both oscillatory and in the limit quadratic. From these analyses, a spline formulation for the basis could be derived allowing a smoothing mechanism to remove the oscillations. By appropriately synchronising and scaling VOS changes in neighbouring cells, this approach was shown to have a dramatic impact on the effectiveness and efficiency of the RSVS on an aerodynamic shape optimisation case.

It was also found that the response of the geometry was discontinuous around topology changes and when constraints go from 0 to non-zero volume fraction requirement. To alleviate the geometric discontinuity around topology changes a progressive refinement approach to the RSVS was presented and shown to automatically generate VOS layouts for geometric inverse design of aerofoils, to a similar tolerance than manually designed grids in previous chapters.

With these developments the RSVS can be used effectively in design optimisation frameworks. The smoothing and refinement methods provide the flexibility for the parameterisation to be used for a wide variety of optimisation problems with limited expertise from a user. In the next chapter the RSVS is integrated into a modular aerodynamic optimisation framework and is used to tackle a range of shape and topology optimisation cases.

CHAPTER 5

FRAMEWORK FOR OPTIMISATION OF TWO DIMENSIONAL GEOMETRIES

Previous sections have developed the restricted snakes volume of solid (RSVS), a combined shape and topology parameterisation method tailored to aerodynamic problems. Chapter 5 presents the integration of the parameterisation into a modular optimisation framework of the type commonly used for aerodynamic shape optimisation (ASO). Figure 5.1 shows the structure of this type of framework as well as the option to use refinement to exploit the progressive approach to design variables developed in the previous chapter.

The optimisation framework developed in this section includes both gradient based and agent based optimisers. This variety of optimisers ensures efficiency on traditional shape optimisation cases as well as sufficient exploration of the design space for cases where topology change is expected. These are coupled to a range of objective functions including geometric error, flow solvers and a finite element (FE) structural solver; the full description of the framework is presented in Section 5.1. This range of objectives allows design optimisation cases with different levels of complexity and computational cost.

Results presented in this chapter serve to validate the effectiveness of the optimisation framework on cases with well known optima: shape optimisation cases, including geometric design and drag minimisations are tackled; along with structural topology optimisation (STO) cases. Drag minimisation of the aerodynamics design optimisation discussion group (ADODG) case 1 benchmark is presented; allowing direct comparison to other aerodynamic shape parameterisation methods [57, 194]. The geometric matching of

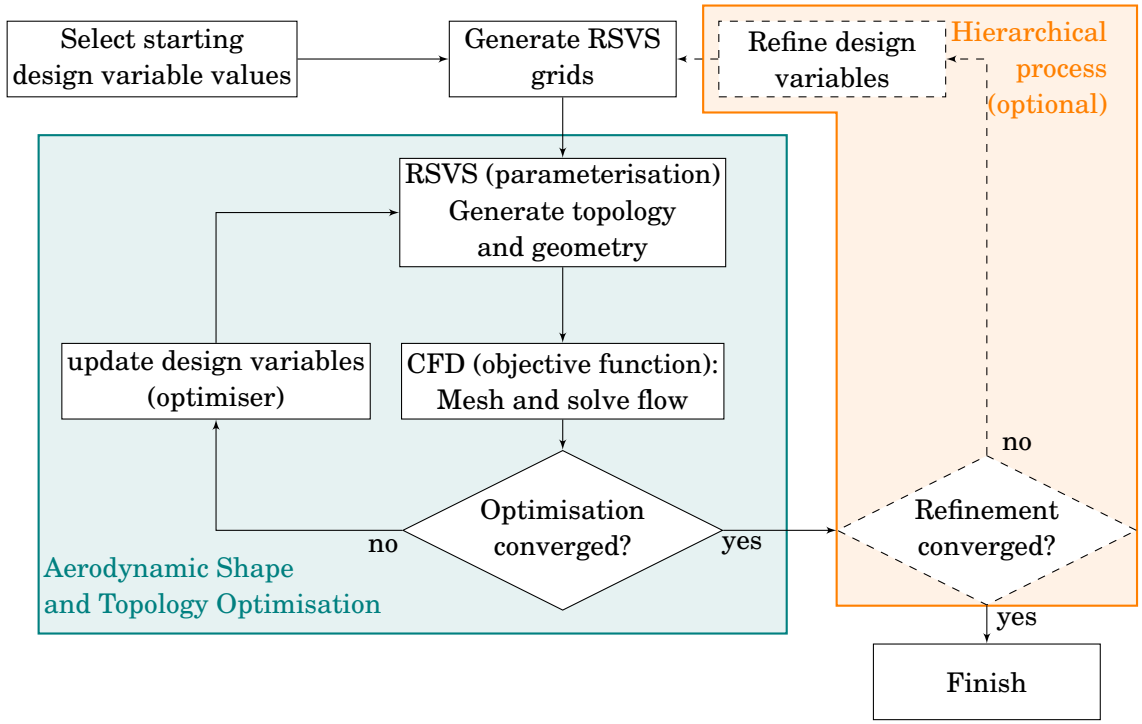


Figure 5.1: Proposed Aerodynamic shape optimisation framework for the RSVS parameterisation.

single and multi-body NACA profiles from the same starting configuration highlight the flexibility of the RSVS to adapt to a problem. Finally, STO cases are used as a validation of the topology optimisation capability.

5.1 Design Optimisation Framework

Optimisation in the ASO community relies on modular frameworks which permits easy isolation and testing of components. These frameworks rely on the successful integration of computational fluid dynamics (CFD), parameterisation and an optimiser into a cohesive framework. This section details the optimisers and objective functions which surround the RSVS in the optimisation framework.

In order to exploit the topological flexibility of the parameterisation all elements of the optimisation method need to support profiles made of an arbitrary number of bodies. For this purpose, the volume of solid (VOS) restricted snake (r-snake) parameterisation method was coupled with a Cut-Cell mesh generator, an unstructured Eulerian flow solver and an optimiser. Cut-cell mesh generators provide the required flexibility with sufficient accuracy at a low computational cost [152, 153]. The flow solver is an inviscid,

compressible unstructured code based on the cell-centred approach by Jameson [239] and following the implementation of Eliasson [240]. The cut-cell mesh generator and flow solver were used in previous studies by Hall et al. [241]. Two optimisation methods were implemented: a conjugate gradient method was used where topology was unlikely to be a factor and Differential Evolution (DE) [73] was used in other cases.

5.1.1 Optimisers

The choice of an optimisation method for a given problem is governed by the expected properties of the design space; in the context of the RSVS this will depend on the layout and number of design variables as well as the type of physical analysis. Two types of optimisation methods are available: local, gradient methods; and, global agent based methods. For cases designed to test the effectiveness of the RSVS compared to shape parameterisation methods, gradient based optimisation are sufficient. When topology is expected to change, non-convex design space exploration is required and a global optimiser is used. These optimisers are presented in the following sections and were tested on the Rosenbrock function.

5.1.1.1 Formulation of the conjugate gradient optimiser

Conjugate Gradient (CG) optimisers are easy to implement and provide adequate convergence behaviour in aerodynamic applications. In CG optimisation the step direction is taken to be the conjugate to all previous step directions, this is done to avoid the sequential normal search directions of steepest descent. The step direction is calculated using Equation 5.1.

$$\Delta \mathbf{x}_{cg}^{t} = -\nabla f\left(\mathbf{x}^{t}\right) + \frac{||\nabla f\left(\mathbf{x}^{t}\right)||^{2}}{||\nabla f\left(\mathbf{x}^{t-1}\right)||^{2}} \Delta \mathbf{x}^{t-1} \tag{5.1}$$

The step direction ($\Delta \mathbf{x}$) is defined by the steepest descent direction, which follows the negative gradient direction ($\nabla f(\mathbf{x})$) and the previous step. Once the step direction is defined a line search is performed to find the minimum along that line. The approach in this work was inspired by back-tracking methods. The algorithm started at a predetermined maximum step length, halving the distance along the line at every step. This approach has the benefit of being parallelisable, minimising wall-time of any calculation.

5.1.1.2 Formulation of a quasi-Newton BFGS optimiser

As an alternative to the conjugate gradient, a step direction can be defined by solving a quadratic approximation to the design space around the current position in the design space. This is achieved using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) approximation of the inverse Hessian of the design space. The BFGS approximation is built solely from first derivatives and achieves quadratic convergence close to the optima. Comparison of its convergence properties was discussed in Section 2.1.2.4 along with other local optimisers.

$$\begin{gathered}
\text{Using shorthand}: \quad \mathbf{p}^t \equiv \Delta\mathbf{x}^t \equiv \mathbf{x}^{t+1} - \mathbf{x}^t \\
\mathbf{q}^t \equiv \nabla f\left(\mathbf{x}^{t+1}\right) - \nabla f\left(\mathbf{x}^t\right) \\
\mathbf{B}_{\mathrm{BFGS}}^{t+1} = \mathbf{B}^t + \left(\frac{1 + \left(\mathbf{q}^t\right)^T \mathbf{B}^t \mathbf{q}^t}{\left(\mathbf{q}^t\right)^T \mathbf{q}^t}\right) \frac{\mathbf{p}^t \left(\mathbf{p}^t\right)^T}{\left(\mathbf{p}^t\right)^T \mathbf{q}^t} - \frac{\mathbf{p}^t \left(\mathbf{q}^t\right)^T \mathbf{B}^t + \mathbf{B}^t \mathbf{q}^t \left(\mathbf{q}^t\right)^T}{\left(\mathbf{q}^t\right)^T \mathbf{p}^t} \\
\Delta\mathbf{x}_{bfgs}^{t+1} = -\mathbf{B}^{t+1} \nabla f\left(\mathbf{x}^{t+1}\right)
\end{gathered} \tag{5.2}$$

The BFGS process maintains an approximation $\mathbf{B}^t$ of the inverse of the Hessian; it gets updated at each iteration using the formula in Equation 5.2. This approximation can then be used in a quasi-Newton algorithm to define the next search direction $\Delta\mathbf{x}_{bfgs}^{t+1}$. The same line-search process that is used for the conjugate gradient was applied to the BFGS step.

5.1.1.3 Formulation of the differential evolution optimiser

Differential Evolution is a heuristic global optimisation method proposed by Storn and Price [73], it was selected due to its robustness and ease of implementation both in series and parallel. Unlike other heuristic methods it requires few internal parameters and has shown good results on a range of applications. It has recently gained traction in the ASO community in the study of multi-modality and in attempts to identify multiple minima through niching [82]. Beyond aerodynamics, differential evolution (DE) has been used to tackle many engineering problems and shown great versatility to tackle a broad range of problems [76]. A number of variants of DE exist; in this work $DE/rand/1$, the original algorithm, is used. This section presents the progression process of the DE algorithm used in this work, the process is summarised in Figure 5.2.

DE drives the solution towards the global optimum by combining members of a population; this process follows three stages: combination, crossover and selection. This process is similar to Genetic Algorithms (GA) with differences in how the crossover

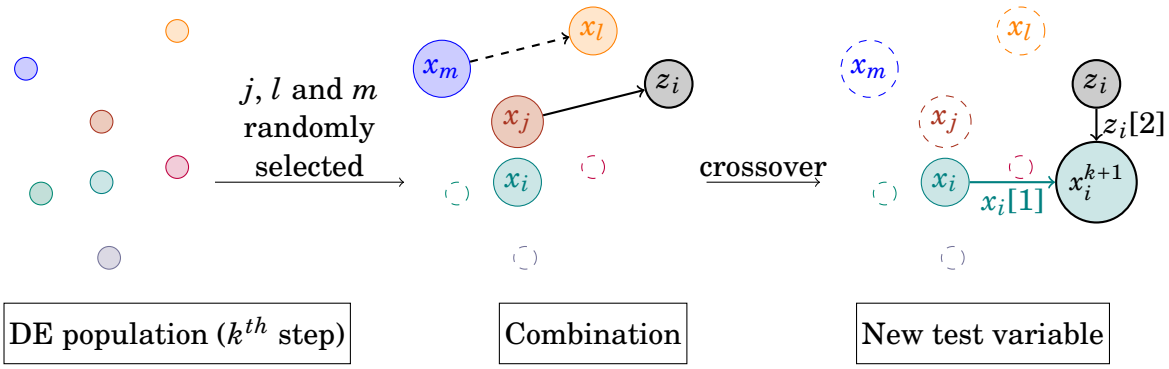


Figure 5.2: Schematic of the DE (DE/RAND/1) process.

and selection stages are carried out. The process is detailed below for the k^{th} step of a differential evolution algorithm with a population of size N in the following paragraphs.

Combination For each member i of the population a random replacement $\mathbf{z}_i$ is generated from a combination of other members according to Equation 5.3. Where $\mathbf{x}$ represents the decision variables of the optimisation process; i, j, l and m are distinct members of the DE population, j, l and m are randomly selected from the population, and, λ is the predetermined *differential amplification factor* (0.5 was used in this work).

$$\mathbf{z}_i = \mathbf{x}_j^k + \lambda\left(\mathbf{x}_l^k - \mathbf{x}_m^k\right) \tag{5.3}$$

Crossover Once the mutated vector $\mathbf{z}_i$ has been generated it is crossed over with the existing member of the population $\mathbf{x}_i$. The components of the vectors to be crossed-over are selected randomly: each component is assigned a number between 0 and 1, this is then compared to the *crossover ratio* σ, which defines whether the component carried through to the next iteration comes from $\mathbf{z}_i$ or $\mathbf{x}_i$. This process is presented in Algorithm 4.

Selection The final Step of the DE process is the selection, this is done using the greedy criterion. The objective value is evaluated at all $\mathbf{x}_i^{k+1}$ which is then compared to the objective function values at the previous time step. Each member of the population is compared to that at the previous step and the one with the best value of the objective function is kept for the next combination and crossover stages. In the case of a minimisation this process is presented in Algorithm 5.

Algorithm 4 Differential Evolution Crossover Process

for Each member of the population; $i = 1,...,N$ **do**
  for Each component of $\mathbf{x}_i$; $j = 1,..,\text{size}(\mathbf{x})$ **do**
    Generate a random number $y_{i,j} \in [0,1]$ with an equi-probable distribution
    if $y_{i,j} < \sigma$ **then**
      Select the component from the mutated vector: $\mathbf{x}_{i,j}^{k+1} = \mathbf{z}_{i,j}$
    else if $y_{i,j} \geq \sigma$ **then**
      Select the component from the original vector: $\mathbf{x}_{i,j}^{k+1} = \mathbf{x}_{i,j}^{k}$
    end if
  end for
end for

Algorithm 5 Differential Evolution Selection process

for Each member of the population; $i = 1,...,N$ **do**
  Evaluate the objective function f at $\mathbf{x}_i^{k+1}$
  if $f\left(\mathbf{x}_i^k\right) \leq f\left(\mathbf{x}_i^{k+1}\right)$ **then**
    Select the member from step k: $\mathbf{x}_i^{k+1} = \mathbf{x}_i^k$
  else if $f\left(\mathbf{x}_i^k\right) > f\left(\mathbf{x}_i^{k+1}\right)$ **then**
    Select the member from step $k+1$: $\mathbf{x}_i^{k+1} = \mathbf{x}_i^{k+1}$
  end if
end for

5.1.1.4 Constraint handling

Most engineering optimisation problems use constraints to represent system wide requirements and ensure a meaningful optimum is discovered. Volume and lift constraints are common in ASO as they avoid degeneracy of drag minimisation problems. Thanks to the VOS formulation of the RSVS volume constraints are easily applied as linear constraints on the design variables. These constraints are easier to handle for the optimiser than the quadratic constraints that would usually be required for other parameterisations.

Inequality volume constraints on the DE optimiser are applied using a "hard-stop" on the design variable values. Other constraints on the DE optimiser are applied using a penalisation barrier method.

In testing the quasi-Newton method using the BFGS approximation, it was found to exhibit worse convergence performance than the conjugate gradient optimiser on aerodynamic problems. It is suspected that aerodynamic sensitivities obtained using finite differences from the flow solver were the cause for this unexpected gap in performance; there is evidence to suggest that conjugate gradient algorithms are more tolerant of

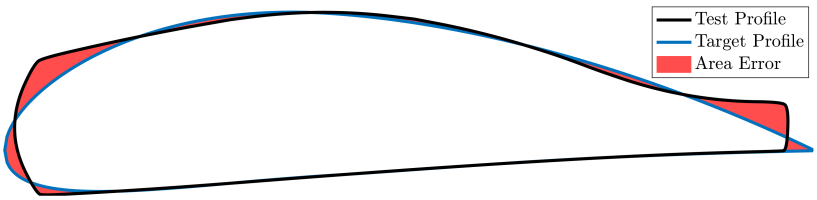


Figure 5.3: Polygons for area error calculation.

lower accuracy gradients. As a consequence many of the gradient based optimisation are carried out using the conjugate gradient method which does not natively support constrained optimisation. Linear equality constraints on the design variables were applied directly at the parameterisation stage; inequality constraints were applied by applying gradient penalisation. Gradient penalisation has the effect of removing the constraint violating search direction without affecting the objective function.

5.1.2 Geometric Matching Metrics

To test the performance of the parametrisation method on different optimisation cases the framework includes two objective functions. The first is a geometric error function which is used to perform geometric inverse design optimisations. The geometric error objective provides a fast and relatively inexpensive test case to evaluate the capability of the parameterisation method. This allows rapid evaluation of changes to the parameterisation without running a set of aerodynamic test cases. It was used to fine tune the smoothing and refinement conditions.

The traditional way of comparing to geometries is to use the distance between two profiles. This works well for similar shapes and can be used to reliably compare the distance between two aerofoils. However to quantify the differences between profiles which may have different point distribution and even different topologies another metric is required. In this work, the geometric error is calculated as the area of the polygons created by intersecting a test geometry and a target profile. For two intersecting geometries, successive intersection points are used to generate closed polygons, for which the area can be computed; their sum is used as an indication of the difference between the curves. This error calculation allows the comparison of curves with different surface point distributions. The normal distance between profiles is also calculated for final profiles to compare the result of this geometric matching to Kulfan's Wind Tunnel Tolerance (Kulfan's WTT).

Notably, the area error shown in Figure 5.3 is used to perform the geometric inverse

design of multi-body aerofoils. In order for the optimiser to prioritise the regions where area is highest, the error is measured as the sum of squared area of each polygon resulting from the intersection of the target profile and the current estimate. Squaring the area has the effect of giving a higher weight to any large polygon. For this objective function the design variables need to be scaled to have the same area response using Equation 4.45.

5.1.3 CFD for Arbitrary Topology

To integrate CFD, optimiser and parameterisation, the RSVS method was coupled with a surface-exact cut-cell mesh generator, an unstructured Eulerian flow solver and an optimiser. In order to exploit the topological flexibility of the parameterisation all elements of the optimisation method need to support profiles made of an arbitrary number of bodies. Cut-cell mesh generators provide the required flexibility with sufficient accuracy at a low computational cost [152, 153]. The flow solver is an inviscid, compressible unstructured code based on the cell-centred approach of Jameson [239] and following the implementation of Eliasson [240]. The cut-cell mesh generator and flow solver were used in previous studies by Hall et al. [241]. A mesh convergence study was performed on the zero-lift drag coefficient value for NACA 0012 at a Mach number of 0.85, giving 469.1 drag counts, which is within 0.3 counts of previous studies using different solvers [176, 189].

Because the RSVS uses traditional boundary fitted meshes it can be used with RANS solvers or any other physical solvers which uses that type of mesh. The main challenge to using the RSVS with viscous CFD is the generation of a suitable mesh without a priori knowledge of the topology. However unsupervised automatic mesh generation for viscous layers has been an active area of research recently seeing implementation in industrial codes [242]. While the current framework does not implement a viscous solver, the study of the parameterisation and its performance on benchmark problems are sufficient predictors of its suitability in those other frameworks.

In the current framework the gradients were obtained by central difference on the flow solution. While unstructured adjoint solvers exist SU2 does not support the arbitrary cells required for cut-cell meshes, and CART3D used by Anderson et al. [154] was not available for licensing with the adjoint solver. The ease of implementation and parallelisation of finite differences made it a suitable option for the test cases considered in this thesis. To ensure consistent discretisation error during the finite difference process, mesh motion is needed. Mesh motion was carried out using the multi-scale RBF

Table 5.1: Validation and mesh convergence of the CFD process on the NACA 0012 at Mach 0.85

Refinement Level	Cell Height (% chord)	C_D	Symmetric Half Mesh Actual Cell Count	Full Cell Count
11	1.221%	525.31	7332	14664
12	0.610%	480.25	13729	27458
13	0.305%	471.48	25276	50552
14	0.153%	469.16	36211	72422
15	0.076%	468.68	57799	115598
16	0.038%	468.56	100512	201024

algorithm of Kedward et al. [243] as it allows efficient and exact movement of large meshes.

5.2 Surface Point Distribution Control

Control of the distribution of surface vertices is critical to the generation of good quality surface meshes and to carry out accurate inverse design. Unusually for a parameterisation method, the RSVS does not automatically yield an analytical form which can be sampled arbitrarily to define a smooth geometry. The sampling rate of the contour is not explicitly controlled when using a restricted snake: it is defined by the position along the underlying grid. This can lead to sampling rates not suitable for aerodynamic simulations.

This issue is resolved by using a re-sampling scheme based on subdivision surfaces and cubic-spline interpolation. This builds on the research of Masters et al. [176] into

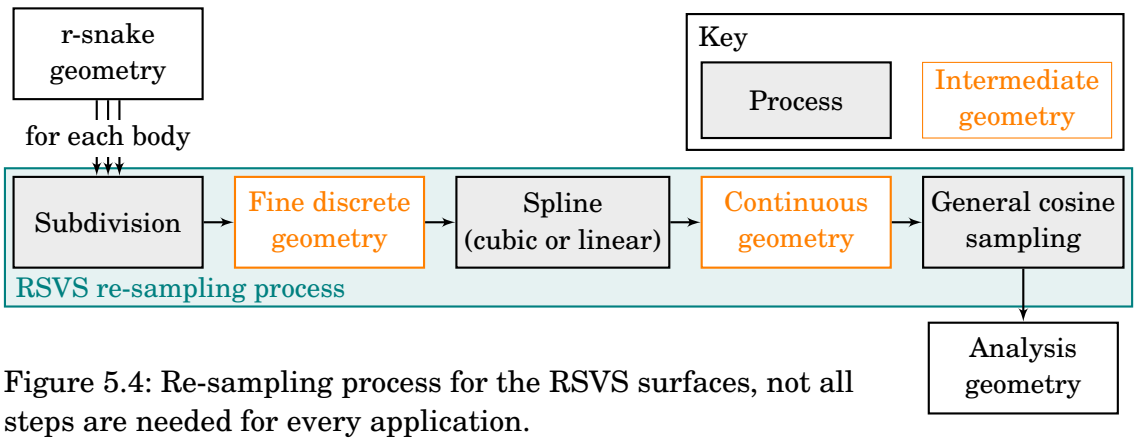


Figure 5.4: Re-sampling process for the RSVS surfaces, not all steps are needed for every application.

subdivision curves for aerodynamic parameterisation. The first step is to refine and smooth the RSVS profile using subdivision curves, then project the resulting points to the limit surface. Subsequently this finely sampled geometry is interpolated using splines and the profile is re-sampled to match the requirements of the application. The process is summarised in Figure 5.4.

5.2.1 Subdivision for Smooth Surface Point Distributions

A range of subdivision schemes have been developed by the computer graphics community to generate smooth shapes from a small number of control points. Chaikin and B-splines subdivision curves have successfully been used in aerodynamic parametrisation by Masters et al. [176]. Subdivision schemes rely on the sequential linear combination of control points to create a new, larger, set of points with pre-determined properties. The ability of these methods to deal efficiently with an arbitrary set of points makes them ideal for the refinement of profiles generated by the snaking process. The general formulation of subdivision schemes is presented in Equation 5.4.

$$\mathbf{p}_{k+1} = \mathbf{S}_k \mathbf{p}_k \tag{5.4}$$

Where $\mathbf{p}_k$ are a set of points transformed by matrix $\mathbf{S}_k$ into a larger set of points $\mathbf{p}_{k+1}$. $\mathbf{S}_k$ is a matrix built from repeating a sub-matrix known as the mask. The mask governs the properties of the vertex averaging process and can vary depending on the goal of the application. The methods named above are primarily designed to generate smooth shapes, but other properties can be designed into the process to suit other applications. Figure 5.5 shows 4 steps of the refinement process for a parabolic smoothing operator. To leverage the maximum potential of Volume of Solid parametrisation and subdivision curves the development of a subdivision process conserving internal area is explored in Appendix B.

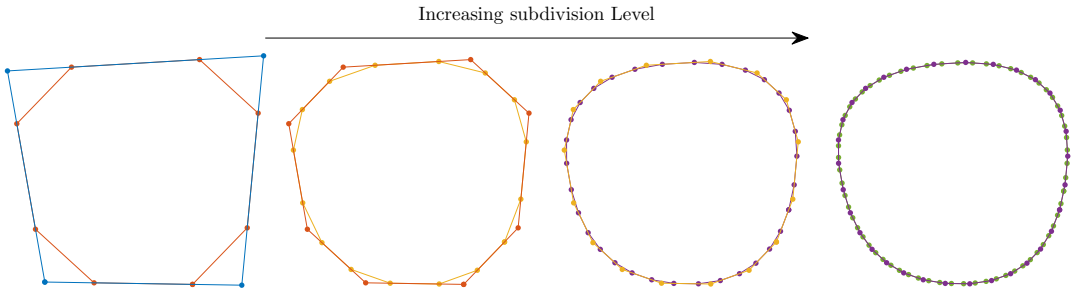


Figure 5.5: Four levels of subdivision of a four point control polygon.

The subdivision re-sampling process outlined above does not enforce the area constraint which is used as the design variables: it tends to generate a profile with a slightly smaller internal area. While this does not affect the quality of the control of the overall algorithm: it remains smooth and reliable; it does affect the ability to directly impose area constraints on the parameterisation. To maintain this desirable property a constant area subdivision process is developed in Appendix B but was not used in optimisation as the differences were found to have little impact on optimisation results. The discovery of the analytical, explicit form of the RSVS limited the impact of such a subdivision scheme as a smooth exact limit curve described by the RSVS can be computed by other means.

Previous work using subdivision curves, to parameterise and re-sample aerodynamic geometries have only needed to generate aerofoils in 2-dimensions; this allowed the use of the well known "cosine distribution" to determine the correct point locations. In topology generation there is no guarantee that an object will have a shape compatible with standard point distributions. For a robust re-sampling of arbitrary topologies a generic equivalent is developed in the next Section 5.2.2.

5.2.2 General Condition for Smooth Surface Sampling of 2-Dimensional Geometries

In aerodynamics the generation of a high quality domain discretisation and a subsequent accurate flow solution is dependent on effective surface meshing. The works by Masters et al. [9, 57] showed the need for consistent surface meshing to avoid arbitrary skews on optimisation results. Aerodynamic parameterisation methods can usually rely on the well understood nature of the designs they represent: for aerofoils there are custom distributions which have been shown to be effective. In three dimensions the main methods are deformative, meaning that the quality of an existing point distribution needs to be maintained rather than a new one generated.

In two dimensions the most common point distribution is the cosine distribution. Following the definition by Masters et al. [57], the trailing edge is a sharp point and the leading edge is the furthest point from the trailing edge. The upper and lower surfaces are sampled independently at regular intervals between 0 and 1 in a parameter space which maps onto the x-axis using Equation 5.5.

The cosine distribution leads to efficient meshes as it has a low element size in regions where the flow changes rapidly (at the leading and trailing edges) and fewer points in between. However this distribution is limited to profiles which have a well defined

$$x_i = \frac{1}{2}\left(1 - \cos\left(\frac{i}{N}\pi\right)\right) \text{ for}: \quad \forall i \in \{0, 1, \cdots, N\} \tag{5.5}$$

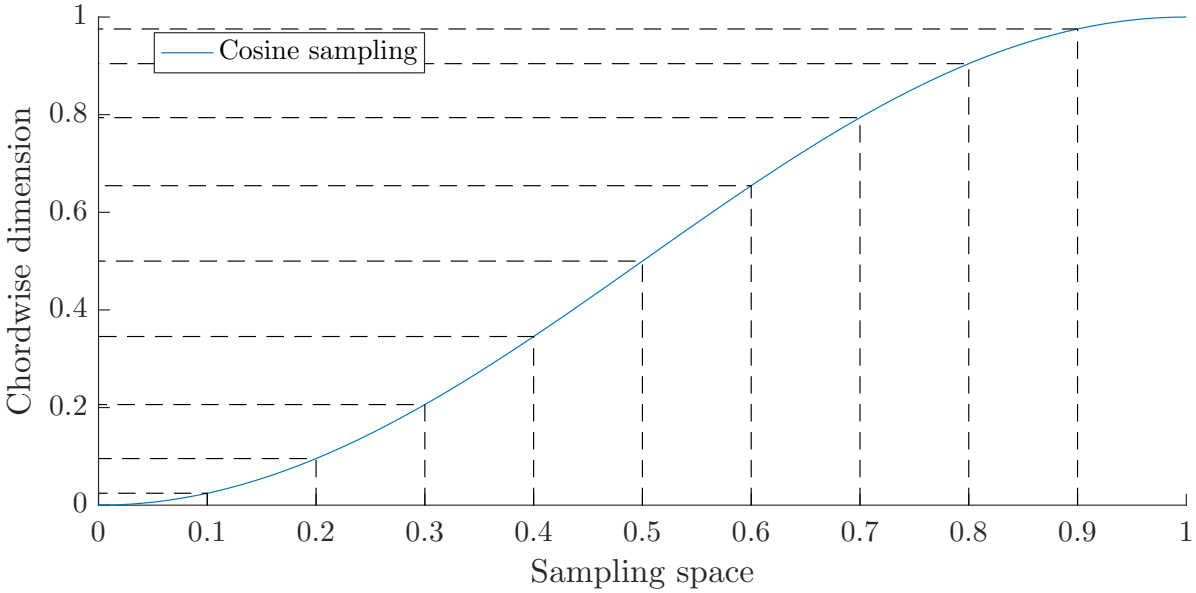


Figure 5.6: Cosine re-sampling following Equation 5.5.

number of surfaces and are monotonic in x. Not all profiles generated by the RSVS have these properties, and some exhibit significant concavities. For this reason, a more general heuristic for achieving fine surface sampling of curved regions and low sampling rates in regions with little aerodynamic impact must be developed. Other concerns in the specification of this rule is to ensure no artefacts are generated regardless of the quality of the original sampling. Through early investigations it was found that it was desirable to have symmetrical sampling around corners.

The re-sampling of profiles of arbitrary topology is done closed contour by contour. To avoid a strong directional dependency, both x and y coordinates coming out of the RSVS process are parameterised as a function of the (discrete) arc-length along the profile. The re-sampling process must generate clustering of points around regions of high curvature in this arc-length parameter space. “Points of high curvature” are identified based on a triangular moving average of the absolute value of the angle between the directions of neighbouring segments (measured in $[-\pi, \pi]$). The two largest values of this averaged distribution are retained as “points of high curvature” along with any points which have a value superior to a quarter of the smallest of these two points.

These points act as the regions of higher clustering, like was the case at 0 and 1 in the cosine distribution (Figure 5.6). Centred around each of these points is the central

$$e_i = \begin{cases} \frac{1}{2}\left(-1 + \frac{\cos\left(-\frac{i-N}{2N}\left(\cos^{-1}(-\varepsilon) - \cos^{-1}(\varepsilon)\right)\right)}{\varepsilon}\right) & \text{for}: i < N \\ \frac{1}{2}\left(1 - \frac{\cos\left(\frac{i-N}{2N}\left(\cos^{-1}(-\varepsilon) - \cos^{-1}(\varepsilon)\right)\right)}{\varepsilon}\right) & \text{for}: i \geq N \end{cases} \quad \forall\ i \in \{0, 1, \cdots, 2N\} \tag{5.6}$$

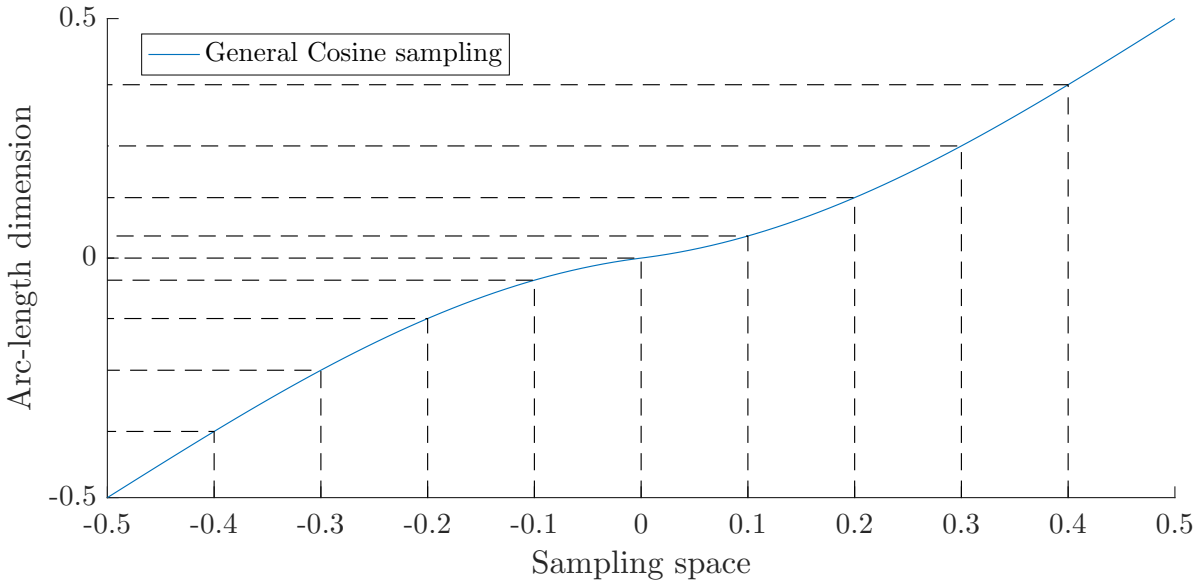


Figure 5.7: General cosine sampling rule following Equation 5.6 with $\varepsilon = 0.98$, a point requiring a high sampling rate is place at the origin of this distribution.

cosine distribution presented in Equation 5.6 and plotted in Figure 5.7. This distribution was used in all subsequent work with a value of ε of 0.98 which allows for sufficient clustering around points of high curvature.

To build a custom distribution adapted to a given geometry a number of the general cosine samples need to be scaled, shifted and assembled. For each "point of high curvature" ($p_{C,j}$) one general cosine patch is centred on it. The cosine patch is then scaled to stretch to the mid-point to the furthest neighbour; the distribution is truncated at the mid-point to the nearest neighbour. This process leads to a smooth distribution in areas of high curvature and a very slight discontinuity where the patches meet in regions of the profile which have been identified as of low importance to the surface meshing process. Each patch is scaled to occupy the same portion of the parameter space as the original did ensuring that no region is under sampled.

This entire process is represented graphically in Figures 5.8 and Figure 5.9 for an arbitrary geometry and for the NACA 4412 aerofoil respectively. It is important to note

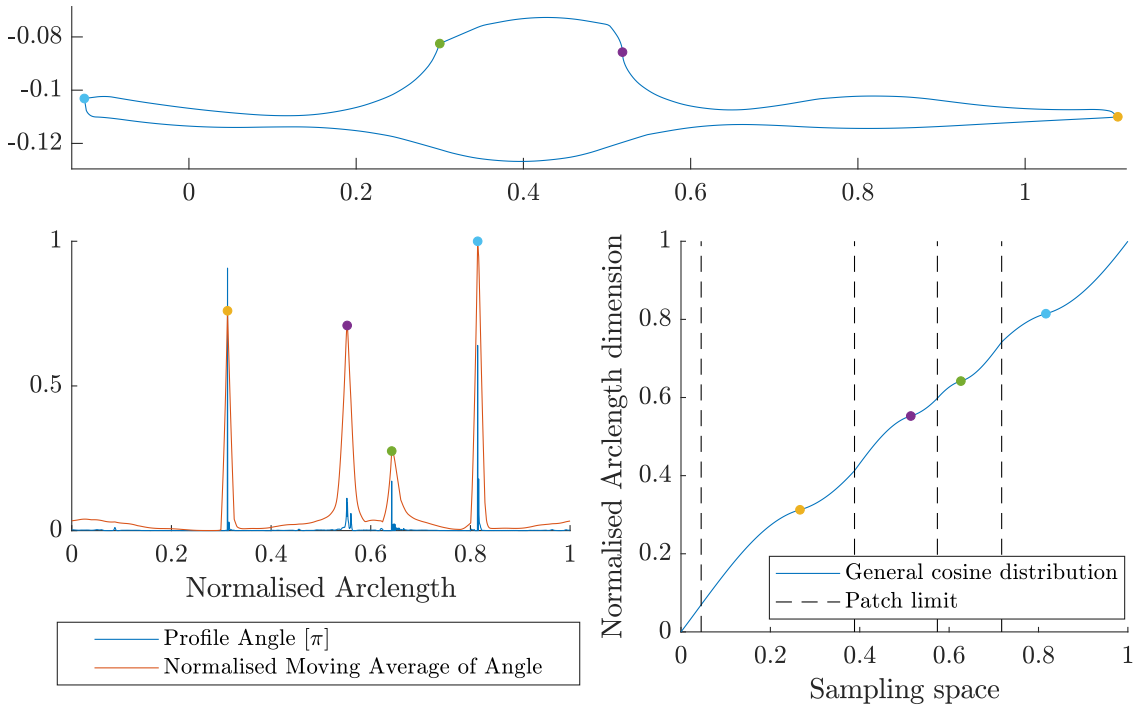


Figure 5.8: General cosine applied to an arbitrary non-aerofoil shape (dots indicate points of high curvature splitting the distribution detected automatically).

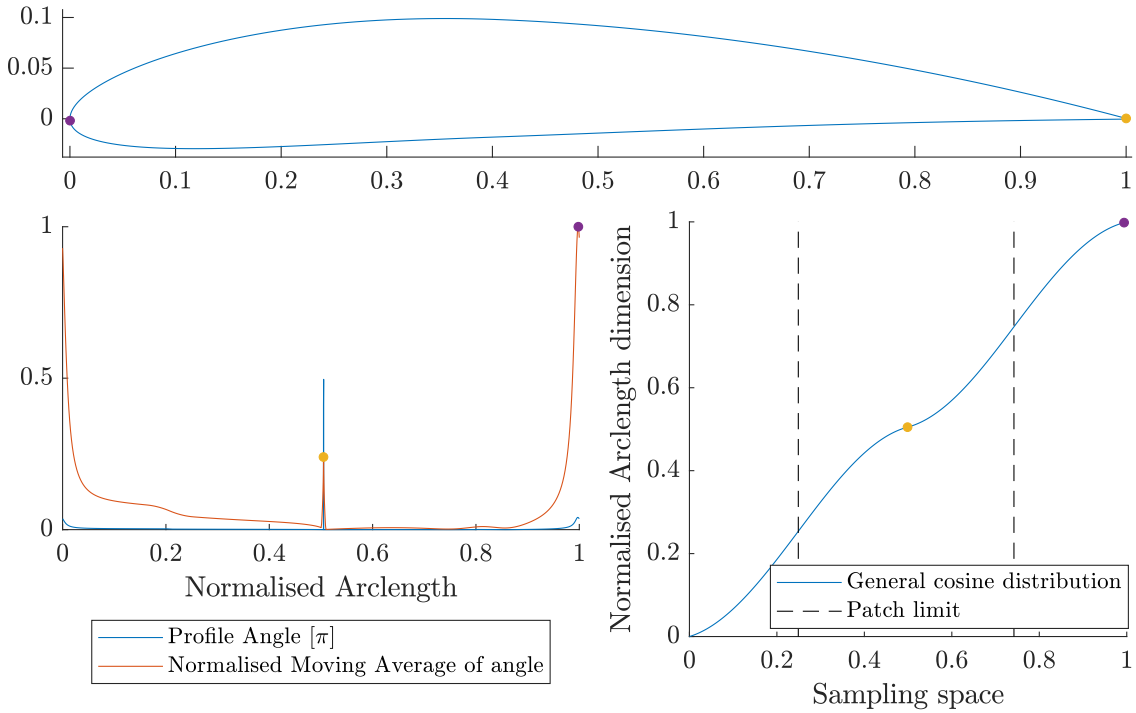


Figure 5.9: General cosine applied to the NACA 4412, sampling is very similar to a normal cosine sampling (dots indicate points of high curvature splitting the distribution detected automatically).

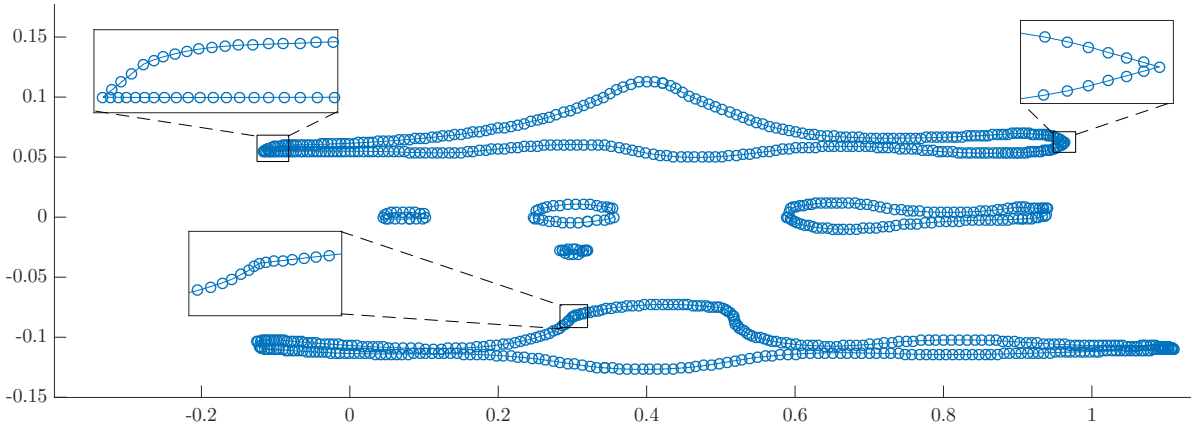


Figure 5.10: Discretised multi-body profile re-sampled using the general cosine rule.

that for the aerofoil like profile leading and trailing edges are correctly identified and the sampling is very similar to a standard cosine distribution. Figure 5.10 shows the entire point distribution generated by the algorithm outlined in this section for a multi-body profile. All the sharp points are correctly identified and densely sampled; flat regions have longer elements allowing efficient meshes to be generated. This algorithm allowed the generation of high quality triangular meshes used in Chapter 5.

5.3 Validation of the Shape and Topology Optimisation Framework

In this section the RSVS is tested in the optimisation framework on aerodynamic shape optimisation problems, the cases tackled show the effectiveness of the parameterisation on problems with known solutions. The goal is to show how simple layouts of design variables can be effective for optimisation. The following tests were performed:

- **ADODG NACA0012 benchmark case 1:** The framework is first tested on benchmark case 1 of the ADODG, an inviscid, transonic optimisation of the NACA 0012 aerofoil. This case is a very useful optimisation benchmark as all established parameterisations were tested on this problem by Masters et al. [9].

- **Geometric inverse design of NACA profiles:** to validate the development of the parameterisation, design variable smoothing and refinement, geometric inverse design cases were used as a cheap alternative to full CFD based optimisation. This group of problems requires similarly smooth and well behaved design variables

as seen in previous studies of aerofoil parameterisation by Masters et al. [9] and Castonguay and Nadarajah [164]. These inverse design cases are unconstrained minimisations of the area error objective described in sub-Section 5.3.2. A range of aerofoils were tested using this process including the NACA4412 and multi-body aerofoils assembled from NACA profiles.

- **STO with the RSVS[1]:** while the inverse design cases presented in this section provide some indication of the topological control enabled by the RSVS, structural design problems provide a more comprehensive test bed for the topological optimisation framework. The development of the RSVS framework for STO was done by Alex Taylor for his Masters research project, and led to a joint publication. It is presented here as it validates the use of RSVS and DE for topology exploration.

5.3.1 NACA 0012 Under Local Thickness Constraints (ADODG Case 1)

To benchmark the optimisation framework and validate the RSVS method the ADODG case 1 was modelled [194]. Case 1 has been extensively studied by a large number of research groups in the ASO community with a wide range of frameworks [154, 169, 176, 183–192]. This makes it a very useful tool in evaluating the competitiveness of the RSVS as a parameterisation for ASO applications. The ADODG case 1 deals with the drag minimisation of the NACA0012 aerofoil, under the constraint that the final profile must lie outside of the NACA0012. The formulation of this optimisation problem is presented in Equation 5.7; the constraint is a localised thickness constraint at every point along the aerofoil chord.

$$\begin{aligned} \min: \quad & C_D \\ \text{s.t.}: \quad & y_{PROFILE}(x) \geq y_{NACA0012}(x) \quad \forall x \in [0,\ 1] \\ & M = 0.85 \end{aligned} \tag{5.7}$$

Work in the ASO community has highlighted many of the challenges associated with the aerodynamics of this case. These difficulties include: premature convergence on sub-optimal profiles [186], asymmetric flow solutions for symmetric profiles [168, 186], oscillatory CFD solutions [244] and hysteretic behaviours with Mach number [188]. The range of observed behaviours was reviewed by Destarac et al. [194].

This case was tackled with the CG optimiser with the mesh resolution at the 14^{th} refinement level which corresponds to a cell height of 0.153% of chord. This is equivalent

Table 5.2: Drag results (counts) for ADODG NACA 0012 benchmark case.

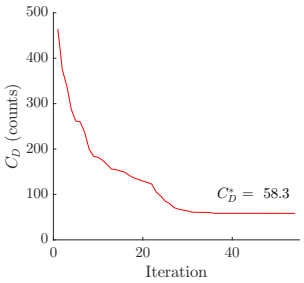
ADODG Case 1	Optimisation Mesh
Cell height	0.153%
Optimised Profile (drag counts)	58.3
NACA 0012 (drag counts)	469.2
ΔC_D (counts)	-410.9
Relative Change	-87.57%

to approximately 1300 cells uniformly distributed around the aerofoil profile. To avoid poor optimiser convergence due to asymmetric flow solutions on symmetric profiles, the optimisation was run on a half mesh with a symmetric boundary. Mesh-motion was performed using the multi-scale radial basis function method developed by Kedward et al. [243]. Constraints were enforced directly on the volume fractions, constraining them to be larger than required for the inverse design of the NACA 0012 aerofoil. While this does lead to a slight constraint violation in front of the point of maximum thickness it is sufficiently precise to capture the complexity of the problem. A 12 by 2 VOS grid was used: half the VOS cells are distributed in a half cosine distribution from leading edge to the 30% chord and until the trailing edge. Symmetric profiles are generated by mirroring the VOS values along the horizontal axis, meaning that the optimiser controls 12 effective design variables.

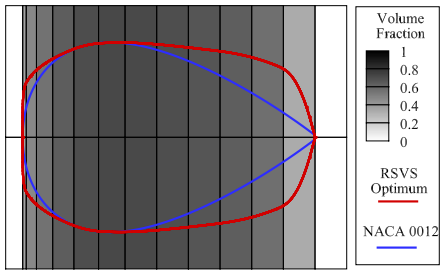
This framework allowed a drag reduction from 469 counts for the NACA 0012 to 58.3 counts for the optimised profile (Table 5.2). This drag reduction is close to the drag values between 50 and 25 counts achieved by other aerodynamic parameterisation methods in recent comparative studies [9, 194]. The shock pattern at the trailing edge of the RSVS optimised aerofoil displays a supersonic/supersonic wave with a single supersonic region over the aerofoil (Figure 5.11). This shock pattern is similar to that observed in the review of this case performed by Destarac et al. [194]. While the drag is not as low as some previous available results, this optimisation case shows that the combination of the RSVS method with the cut-cell mesh generator is capable of exploring a complex aerodynamic design space. Flow features expected in this optimisation case are successfully discovered by the optimisation framework.

Previous research into this case has shown that the optimum lies in a hysteresis loop with regard to Mach number: there is a ‘high’ and ‘low’ drag branches. This means that small perturbations can lead to a large difference in the final drag value: this case is extremely sensitive to internal stepping parameters and the layout of design

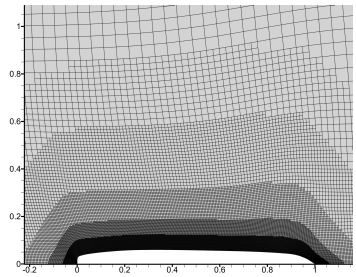
variables. Comparing the profile in Figure 5.12 to those found in previous studies, the angle at the trailing edge would be expected to be wider. This is certainly due to the optimiser converging on a cusp trailing edge and finding the 'low drag' branch of the flow solution [194]. Small disturbances in any of the feasible directions will cause the flow to change topology and jump to a 'high drag' branch where there are two distinct region of supersonic flow. The trailing edge has failed to evolve in the right direction and is not optimal. This discrepancy is likely to be linked to the constraint handling of the



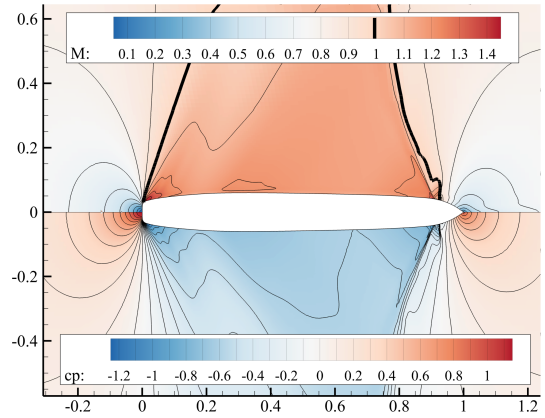
(a) Convergence history of the conjugate gradient optimiser.



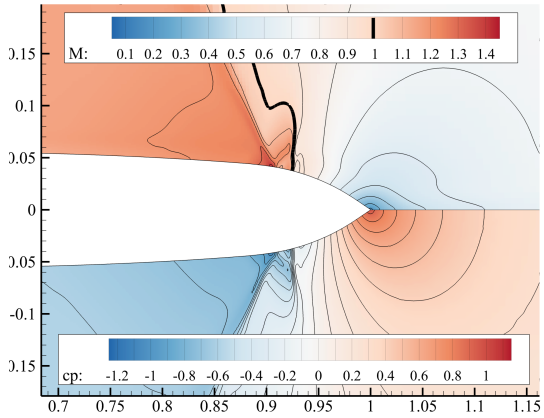
(b) Profiles of the optimum and the NACA 0012 on the RSVS design grid.



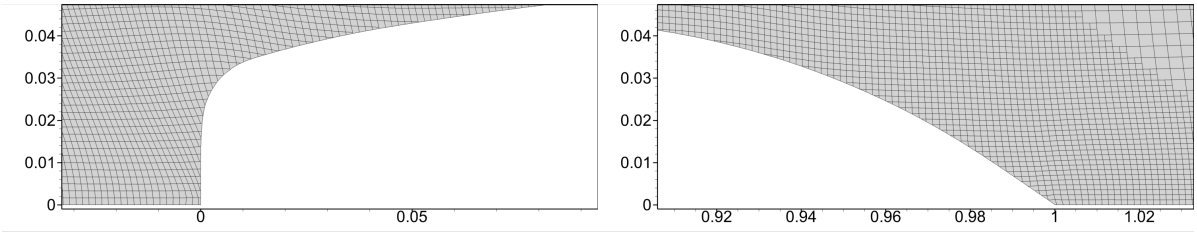
(c) Half-mesh used for the optimisation.



(d) C_P and Mach flood plots for the optimum profile

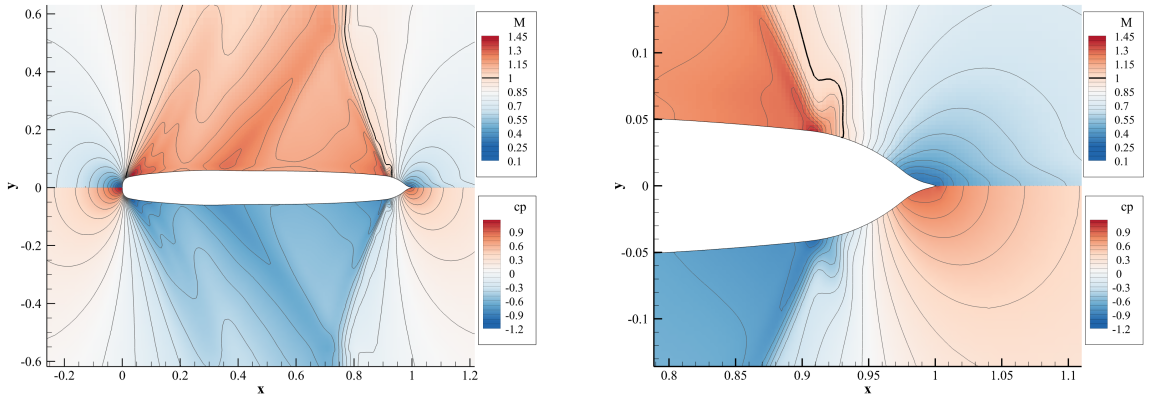


(e) C_P and Mach flood plots at the trailing edge



(f) Detail of the leading and trailing edges of the mesh for the optimised geometry.

Figure 5.11: Optimisation of the ADODG Case 1 using 10 active design variables.



(a) C_P and Mach flood plots for the optimum profile

(b) C_P and Mach flood plots at the trailing edge

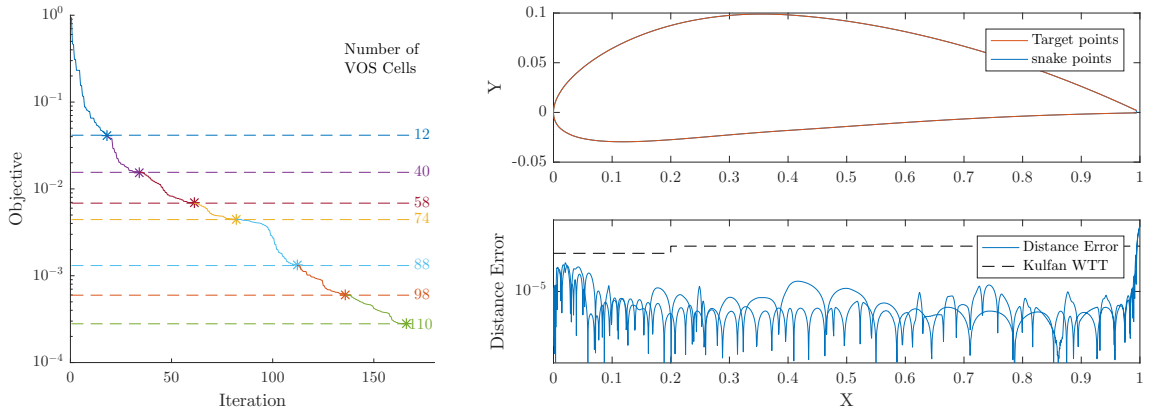
Figure 5.12: Termination of an optimisation run of the ADODG Case 1 using 10 active design variables on a sub-optimal design which generates 66 drag counts.

optimiser or the finite differencing which are weak points of the optimisation framework. The BFGS quasi-Newton optimisation method was tested on this problem but proved less effective than CG in the current framework. This is because this type of second order method requires higher quality gradients than CG [43]. An adjoint solver if available could have been used to improve the robustness of the gradient calculation.

Instead of focusing development on this specific test case general solutions to the issue of optimisation robustness have being considered. Recent investigation of the ADODG NACA0012 case using a hierarchical design variable approach has led to drag values of 4.2 counts [176]. VOS design variable formulations lend themselves well to hierarchical approaches as the grid can be split to increase the resolution or merged to return control to broader design variables. The intuitive approach to refinement allowed by the r-snake VOS parameterisation is one of its main benefits and will allow robust optimisation results.

5.3.2 Geometric Inverse Design with Refinement

The previous section validated the behaviour of the RSVS parameterisation inside a traditional aerodynamic shape optimisation set-up. In order to test the refinement process in the context of optimisation the geometric matching of the NACA 4412 aerofoil is tackled as well as that of a two body aerofoil. Unlike refinement results presented during the development of the refinement process, the cases in this section use the volume fractions as design variables which are driven by the conjugate gradient optimiser. These



(a) Convergence history of the optimisation process

(b) Profile distance between the NACA4412 and the final profile

Figure 5.13: Convergence history and final local profile error for the Geometric Inverse design of the NACA 4412 over 6 refinement steps.

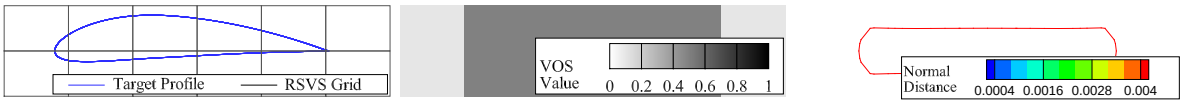
test the ability of refinement to tailor grids to individual cases and the capability of an optimiser to explore those automatically generated layouts of design variables.

5.3.2.1 NACA 4412

The refinement process is first tested on the geometric inverse design of a NACA 4412 aerofoil, using the smoothing described earlier combined with the conjugate gradient optimiser. The goal of this test is to explore the flexibility of the method and the quality of the integration with the optimiser.

This case was tackled with 6 refinement steps starting from a coarse grid of 2 by 6 design variables, the evolution of the objective function is shown in Figure 5.13a, the grid for each refinement step is shown in Figure 5.14 along with the corresponding profile and volume fractions. The set-up of this case was done to test the effectiveness of the RSVS parameterisation with local refinement rather than the capacity of the RSVS to recover a NACA 4412. For this reason the profile was allowed to evolve freely over the grid with no constraint on the position of leading edge and trailing edges.

The final profile (Figure 5.13b) shows the parameterisation successfully built a smooth leading edge and sharp trailing edge. Building sharp trailing edges is straightforward in the RSVS parameterisation, it simply needs very small volume fractions in design cells. However this requires a design grid intersection very close to the desired location of the trailing edge which requires many refinement steps to converge as shown in Figure 5.14h.



(a) Start of refinement step 0 (iteration 1)

(b) End of step 0 (iteration 18)

(c) End of step 1 (iteration 34)

(d) End of step 2 (iteration 61)

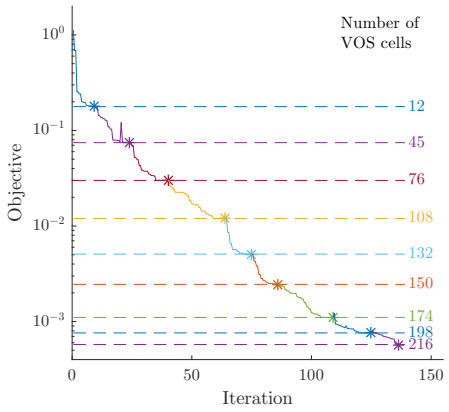
(e) End of step 3 (iteration 82)

(f) End of step 4 (iteration 114)

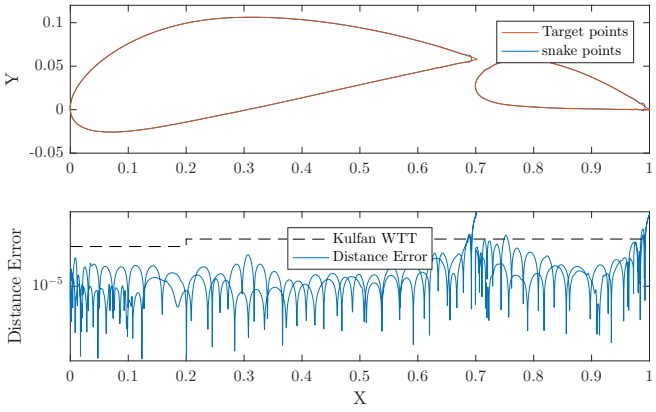
(g) End of step 5 (iteration 136)

(h) End of refinement step 6 (iteration 169)

Figure 5.14: Geometric recovery of a NACA4412 using 6 refinement steps; with the RSVS grid and the target profile (left), the VOS values for the geometry (centre) and the corresponding profile coloured according to its normal distance to the target profile (right) at the first and final iteration.



(a) Convergence history of the optimisation process



(b) Profile distance between the multi body aerofoil and the final profile

Figure 5.15: Convergence history and final local profile error for the Geometric Inverse design of the multi-body aerofoil over 8 refinement steps.

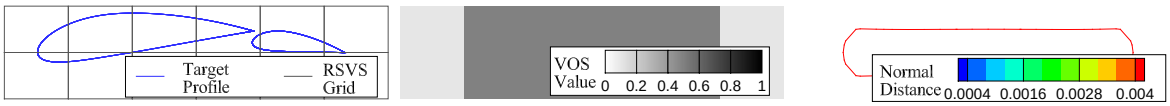
5.3.2.2 Multi-body NACA profile

The same process with the same starting condition but with eight refinement stages was used to tackle the geometric recovery of a multi-body aerofoil composed of 2 NACA 4 digit profiles. Figure 5.15b shows the final profile with the normal distance to the target profile; Figure 5.16 the evolution of the RSVS profile and grid through the optimisation. This second case highlights the versatility of the RSVS with refinement, their combination allows two optimisation cases with very different solutions to be tackled without user intervention.

Figures 5.13a and 5.15a display the convergence behaviour of the optimisations which exhibits a step by step convergence. This is the desired behaviour: as the optimisation converges on a coarse set of design variables, the refinement process selects an appropriate portion of the design variables to refine enabling further reduction of the objective function. The effectiveness of the contour curvature condition (Equation 4.48) is shown by the improvements brought by each refinement level; ineffective selection would lead to stagnation of the objective function.

5.3.2.3 Refinement for other applications

While aerodynamic optimisation benefits from fine geometric control, it is also dependent on global parameters, especially in three dimensions global transformations such as angle of attack, sweep, twist and span must be handled concurrently to the finest



(a) Start of step 0 (iteration 1)

(b) End of refinement step 0 (iteration 9)

(c) End of refinement step 1 (iteration 24)

(d) End of refinement step 2 (iteration 40)

(e) End of refinement step 3 (iteration 64)

(f) End of refinement step 4 (iteration 75)

(g) End of refinement step 5 (iteration 86)

(h) End of refinement step 6 (iteration 109)

(i) End of refinement step 7 (iteration 125)

(j) End of refinement step 8 (iteration 136)

Figure 5.16: Geometric recovery of a multi body aerofoil over 8 refinement steps.

design variables. In the RSVS parameterisation this can be achieved by applying those transformations to the grid so that they are reflected in the profile.

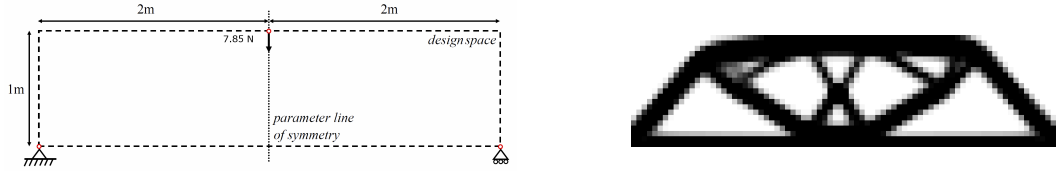
Previous local refinement schemes developed by Masters et al. [176] and Anderson and Aftosmis [154] relied on the adjoint of the objective. Instead of directly using adjoints, information about the objective function is passed indirectly to the RSVS refinement process through the optimisation process through its effect on the shape of the profile. This effect, inherent to any optimisation framework, is coupled with a measure of the 'stress' that is experienced by the parameterisation (here the curvature of the profile) to identify areas where the current parameterisation may not be fine enough. The refinement process of the RSVS relies exclusively on information from the parameterisation and the profiles it generates, this makes this refinement approach portable to any problem tackled using the RSVS parameterisation method with no modification.

5.3.3 STO Using the RSVS Framework

The RSVS was developed as a parameterisation tuned to the specifics requirements of aerodynamic optimisation problems, however steps were taken to avoid restricting it to traditional aerofoil like shape. In an effort to test the RSVS framework on another class of problems, the fully implemented framework was shared with a Masters' student[1], with the goal that a structural objective function would be added to the framework. The introduction of FreeFem++ into the framework was successful enabling structural topology optimisation (STO) cases to be tackled using the RSVS with DE. This work led to the joint conference publication: "Structural Topology Optimisation with R-Snakes Volume of Solid" [124]; details of the implementation and the validation against solid isotropic material with penalisation (SIMP) are available in Taylor et al. [124]. A single case from this publication is presented in this section in order to show the successful topology optimisation of structures; demonstrating the versatility of the RSVS to tackle design optimisation problems beyond aerodynamics.

The case highlighted in this section is the optimisation of the MBB beam, a common STO test case, generated by Alexander Taylor. The case presented here is the minimisation of the deflection of a beam supported at both ends, successful optimisation of this case leads to bridge like structures. The definition of this optimisation case is presented in Figure 5.17. The final RSVS optimised profile is compared to the result of an open

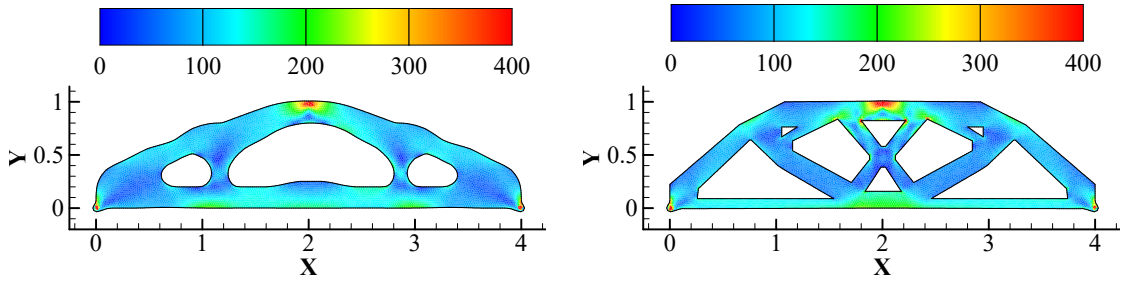
[1] Implementation of the structural component and its validation was done by Alexander Taylor, at the time a Masters student at the university of Bristol in aerospace engineering, under the supervision of C.B. Allen, and the mentorship of the author of this thesis.



(a) Problem definition.

(b) Raw greyscale SIMP solution.

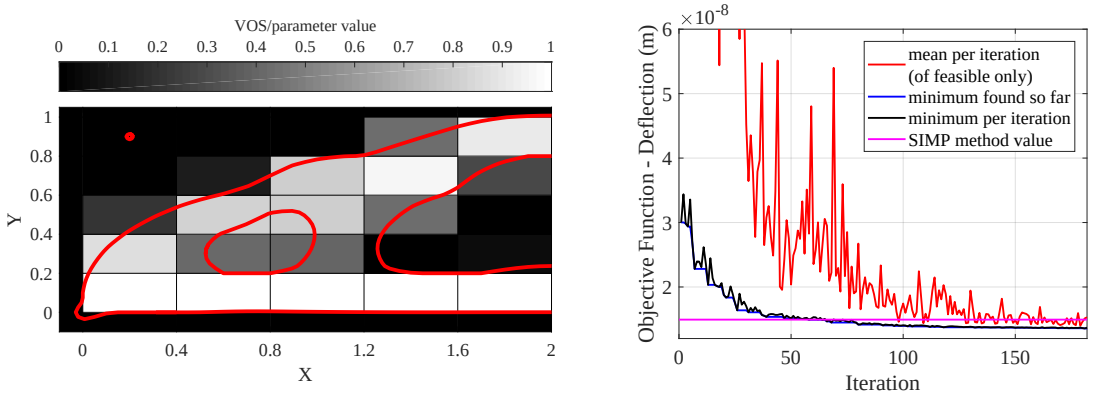
Figure 5.17: Problem definition and raw SIMP solution for the MBB beam (from [124]).



(a) RSVS found solution, volume = 1.995, OF = 1.36e-8.

(b) SIMP smoothed solution, volume = 1.9942, OF = 1.48e-8.

Figure 5.18: Von Mises stress (Nm^{-2}) plots of the optimum RSVS geometry and a benchmark SIMP geometry generated by Sigmund's open source code [125]. The RSVS solution has a 8.1% lower objective function (from [124]).



(a) The 25 parameters of the final solution.

(b) Convergence History.

Figure 5.19: RSVS parameter values and deflection convergence history for optimisation of the MBB beam (from [124]).

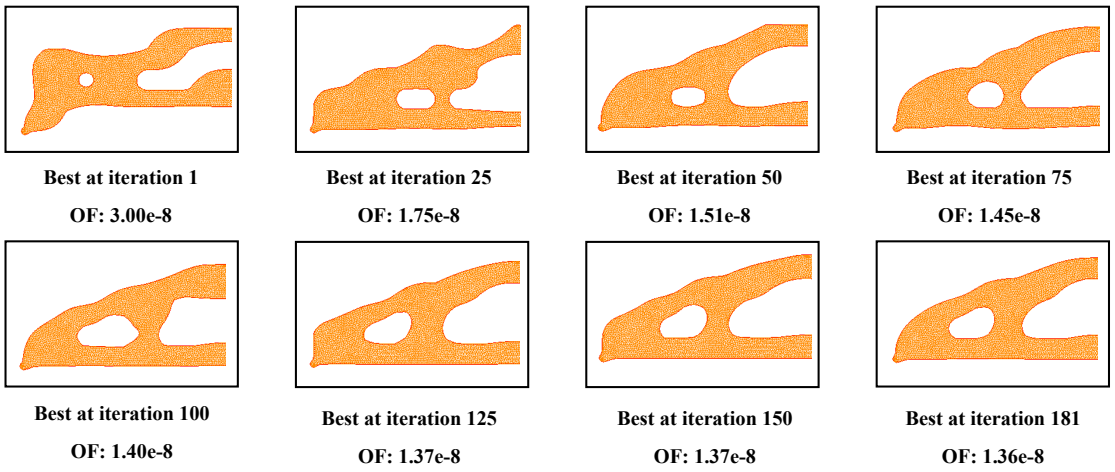


Figure 5.20: Evolution history of the "best so far" geometry for optimisation of the MBB beam using differential evolution (from [124]).

source SIMP process [125] in Figure 5.18, showing the effectiveness of the DE-RSVS optimisation method. Figures 5.19 and 5.20 provide detail of the progression of the optimisation.

The capability of the same framework to tackle aerodynamic and structural topology problems represents a step towards a general method capable of tackling arbitrary design problems. While the current FE method is limited by the absence of adjoint sensitivities and the use of an agent based optimiser, results are promising and do not fully exploit the capability of the RSVS. Instead of needing a bespoke parameterisation and framework for each class of problem, a single modular framework based on RSVS offers the possibility to explore arbitrary topology optimisation problems.

5.4 Summary

The RSVS was integrated into the type of modular optimisation framework common in the ASO community. The optimisation of the ADODG case 1 showed the performance of the RSVS framework to be in line with other established shape optimisation frameworks: the expected aerodynamic behaviours were observed and comparable results were achieved. Shape matching single and multi-body cases showed the ability of the RSVS framework to automatically refine the design variable layout to achieve both topology and shape recovery independent of the starting geometry. The modularity of the framework is further validated by the implementation of a structural objective. These validation efforts indicate that the RSVS framework can find applications on a wide

range of shape and topology design problems.

CHAPTER 6

EFFICIENT EXPLORATION OF AERODYNAMIC TOPOLOGY USING THE RSVS

This chapter explores specific aerodynamic shape and topology optimisation cases in order to measure the impact of topological flexibility on the performance of optimised profiles. The cases in question are the minimisation of drag under area constraints for supersonic conditions presented in Section 6.1. The availability of analytical solutions [232, 245, 246] and previous numerical studies [247] make these useful test cases for the continued validation of the framework. Due to the impact of interacting shock waves on drag, non-aerofoil topologies can have very advantageous properties making supersonic flows a natural regime for topology optimisation.

The cases are tackled using the hierarchical restricted snakes volume of solid (RSVS) and conjugate gradient (CG) optimisers to achieve results comparable to aerodynamic shape optimisation (ASO) frameworks in Section 6.2, then the same optimisations are performed using an expanded design space and differential evolution (DE) in order to explore the impact of topological flexibility on aerodynamic performance. Most of the cases tackled in this work are studied as part of 'families' of cases which require no change in set-up between individual cases but where the results may exhibit very different properties. This explores the ability of the framework to tackle a wide range of cases with limited tuning, to allow case-independent set-up of the optimisation framework.

As part of this effort, the RSVS is integrated into a hybrid parameterisation scheme[1]

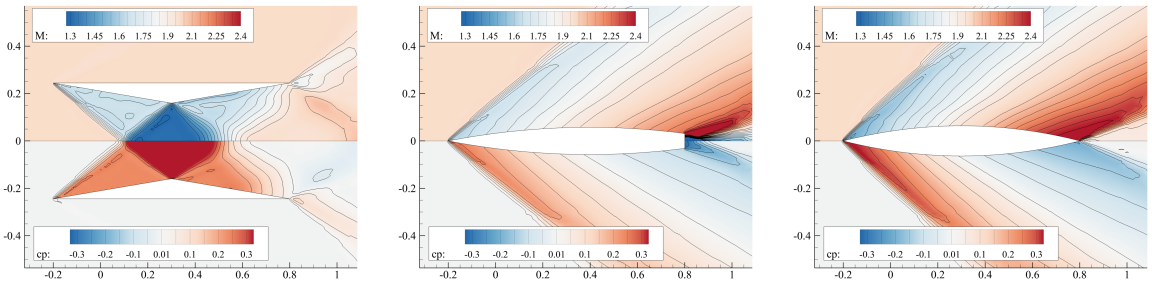
[1] This collaborative work led to the publication of two conference papers: "Efficient Multi-Resolution Approaches for Exploration of External Aerodynamic Shape and Topology" [248], and, "Optimisation of Multi-Modal Aerodynamic Shape and Topology Problems" [249].

which seeks to exploit the topological flexibility of the RSVS and the efficiency and effectiveness of multi-level subdivision optimisation (MLSO) as originally developed by Masters et al. [176]. This work, done in collaboration with Laurence Kedward, aimed to provide the most thorough exploration of the aerodynamic design space to answer questions of geometric and flow multi-modality in the study of aerodynamic topology. Sections 6.4 and 6.5 present the development and use, respectively, of a more effective framework relying on the RSVS for topology control and the MLSO for extremely efficient shape parameterisation.

6.1 Drag Minimisation Under Area Constraints at Mach 2

The drag minimisations in the following sections are inviscid, supersonic, constant area optimisations. Significant research into these cases was carried out in the 1950s using linearised equations for supersonic flow which yielded analytically optimal solutions. In 3 dimensions, this effort led to the now famous Sears-Haack profile for minimum wave drag [245, 250], and the somewhat less famous solution by Klunker and Harder [246] for non-linear supersonic pressure relationships.

Supersonic flows are also an excellent test bed for topology optimisation: there exist multi-plane profiles where shock interactions produce bodies with no wave drag [251]. The most well known of these is the Busemann biplane first proposed in the 1930s by Busemann [232]. These cases are of particular interest as these multi-body profiles can be built using the volume of solid (VOS) parameterisation method: they are known cases



(a) Busemann bi-plane (b) Truncated ogive [246] (c) Ogive

Figure 6.1: Three types of analytical optima at Mach 2 with an area (c_A) of 0.08. Note that each can potentially be the global optimum, depending on the chosen area constraint; for the constraint value of 0.08, the Busemann bi-plane is optimal.

for which topological flexibility should bring significant drag reduction. An example of the flow around each of the three known analytical optima is shown in Figure 6.1.

The mathematical programming representation of these problems are expressed in Equations 6.1 and 6.2, for the single topology case and the multi-body case respectively. The behaviour of the optimisation is dependent on the area constraint value c_A. An additional constraint is added for the multi-body cases to ensure that the optimised profile fits inside the region occupied by the Busemann biplane, the maximum height of the profile (Δy_{max}) cannot be larger than the maximum height of a Busemann biplane ($\Delta y_{BUSEMANN}$). Indeed if the optimisation is allowed to generate profiles far apart it can effectively maintain bodies operating in flows sufficiently separated to be independent of each other.

$$\begin{aligned} \min \quad & C_D \\ \text{s.t.} \quad & \sum \mathbf{a} = c_A \\ & M = 2 \end{aligned} \tag{6.1}$$

$$\begin{aligned} \min \quad & C_D \\ \text{s.t.} \quad & \sum \mathbf{a} = c_A \\ & \Delta y_{max} <= \Delta y_{BUSEMANN} \\ & M = 2 \end{aligned} \tag{6.2}$$

The volume constraints are applied before the parameterisation stage by controlling the values of the volume fractions: if a constraint violation is detected the volume fractions are scaled such that their sum matches the constraint. The next sub-sections show the optimisation of supersonic aerofoils, for a fixed layout of VOS design variables, then using the anisotropic refinement. The impact of the topological flexibility of the parameterisation on these cases is also discussed.

6.2 Single Body Drag Minimisation for a Mach 2 Fixed Area Profile

The first supersonic cases investigated for the current optimisation framework were the drag minimisation of profiles at Mach 2 for a fixed volume and chord using the conjugate gradient optimiser (Equation 6.1). A range of cases were tested for c_A varied from 0.01 to 0.15. The design space was arranged in a 2 by 10 design grid (Figure 6.2c), with

Table 6.1: Drag results (counts) for the thin, constrained area, optimisation.

$\mathbf{c_A = 0.05}$	Optimisation Mesh	Fine Mesh
Optimised Profile	168.72	170.02
Ogive	174.16	174.06
ΔC_D	-5.44	-4.05
Relative Change	-3.12%	-2.32%

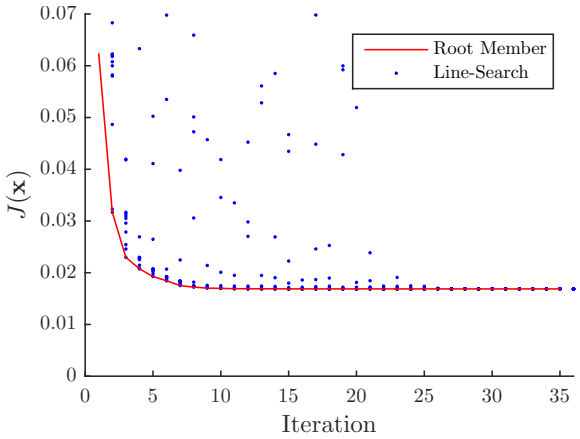
Table 6.2: Drag results (counts) for the thick, constrained area, optimisation.

$\mathbf{c_A = 0.11}$	Optimisation Mesh	Fine Mesh
Optimised Profile	685.61	687.36
Truncated Ogive	687.23	683.45
ΔC_D	-1.62	3.91
Relative Change	-0.24%	0.57%

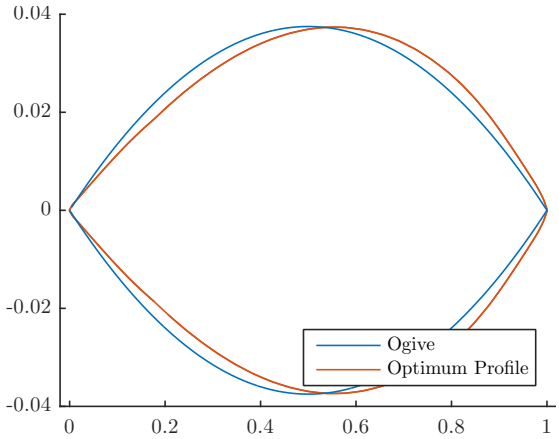
VOS cells following a cosine distribution in x with symmetry of the design variables about the horizontal axis. The chord is fixed by maintaining a small volume fraction in the volume cells at the leading edge and the trailing edge; this effectively limits the parameterisation to aerofoil-like shapes. This set-up is similar to traditional aerofoil parametrisation methods where more control points are clustered towards the leading edge and trailing edges and movements in the vertical direction dominate. This approach uses engineering knowledge to build a suitable grid for the RSVS parameterisation to perform efficiently on the given problem and generates results that are comparable to other existing 2-D optimisation methods.

Figures 6.2 and 6.3 present full results for the optimisation of a thin and a thick profile respectively. Figure 6.4 shows the optimum profiles for each value of the volume constraint. The cases up to areas of 0.08 result in profiles consistent with the parabolic body with one notable feature: as the required volume increases, the point of maximum thickness is shifted towards the trailing edge. This finding is similar to a previous study by Palaniappan and Jameson [247]. This behaviour allows the shock to be weaker for the non-linear optimum than for the corresponding ogive, which more than makes up for the increase in back pressure. The trade-off between back pressure and shock can be seen clearly in the flood plots of thin non-linear optimum (Figure 6.2f) and the parabolic profile (Figure 6.2e). These changes lead to a 2.3% drag reduction (Table 6.1) for the non-linear optimised profile.

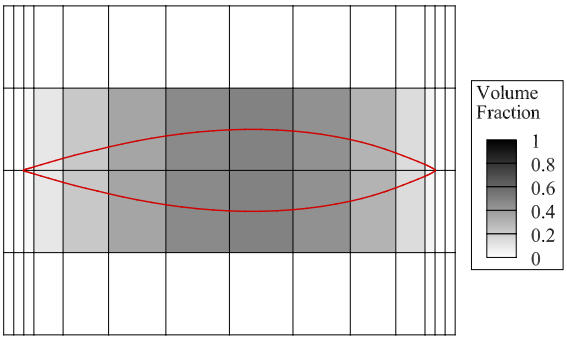
For profiles above an area constraint of 0.08 the profile generated resembles the truncated ogive first defined by Klunker and Harder. Unfortunately the resolution of the design variables at the trailing edge is insufficient to capture the crispness of the truncated trailing edge meaning that the RSVS does not outperform the analytical



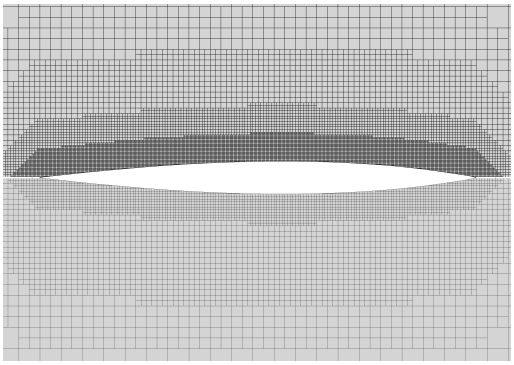
(a) Convergence history of the CG process



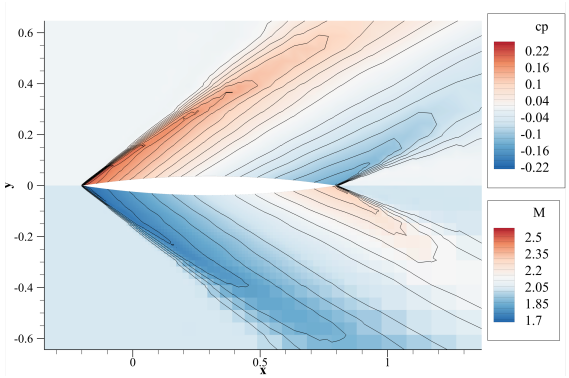
(b) Profiles of the linear and non-linear optima



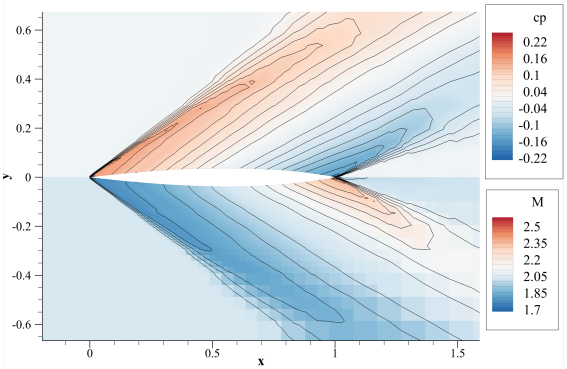
(c) Snaxel parameterisation and contour (Vertical stretch: 2)



(d) Coarse optimisation mesh and fine post-treatment mesh

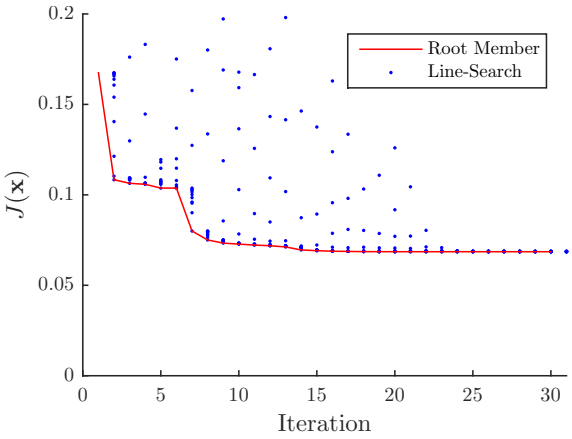


(e) C_P and Mach flood plots of the Parabolic Profile [247]

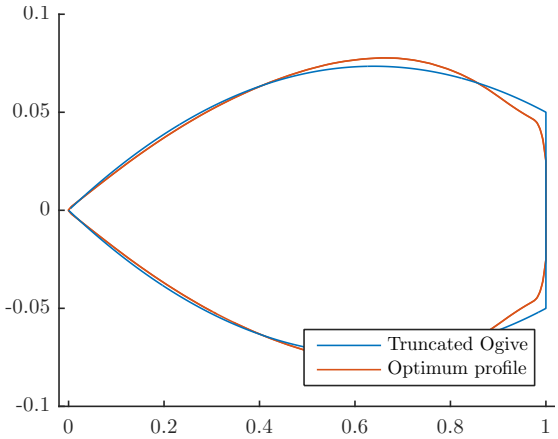


(f) C_P and Mach flood plots for the Non-Linear Optimisation Result

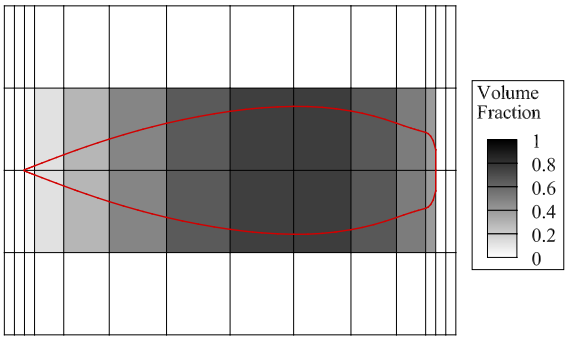
Figure 6.2: Results of the conjugate gradient optimisation for a 2 by 10 design grid with a constrained volume of 0.05.



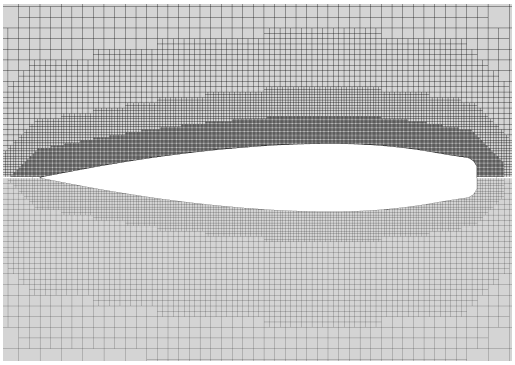
(a) Convergence history of the CG process



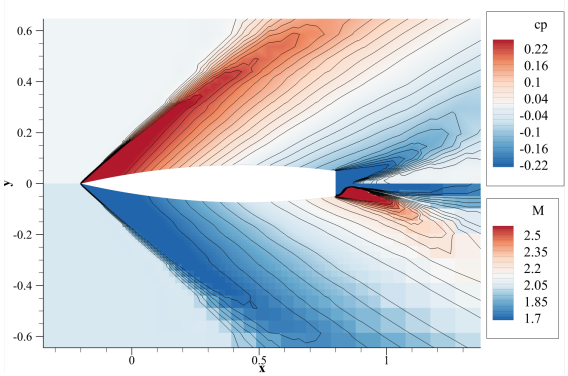
(b) Profiles of the linear and non-linear optima



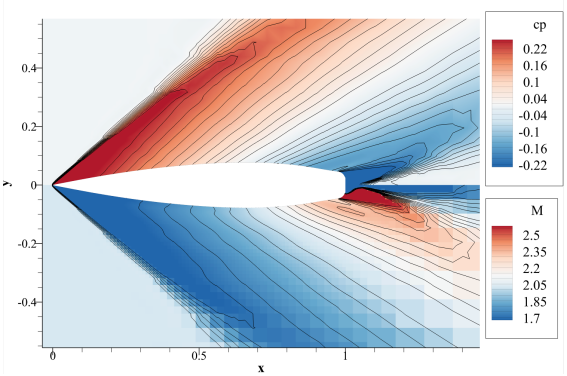
(c) Snaxel parameterisation and contour (vertical stretch: 2)



(d) Coarse optimisation mesh and fine post-treatment mesh



(e) C_P and Mach flood plots of the truncated ogive [246]



(f) C_P and Mach flood plots for the Non-Linear Optimisation Result

Figure 6.3: Results of the conjugate gradient optimisation for a 2 by 10 design grid with a constrained volume of 0.11.

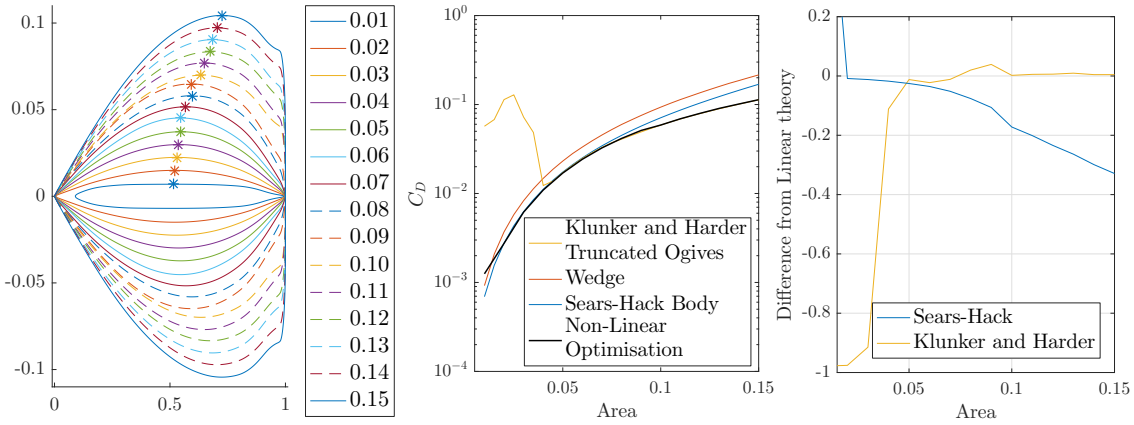


Figure 6.4: Results of optimisation cases for values of area constraint c_A between 0.01 and 0.15, the $*$ indicates the point of maximum thickness.

optima on the finest analysis mesh (0.54% drag increase Table 6.2).

In addition, the optima for constraints above 0.08 present a significant challenge to the flow solver. Indeed the rapid change in angle of the profile at the trailing edge generates a region of separated flow despite the use of an inviscid flow solver. This 'inviscid separation' is due to the artificial dissipation inherent to the numerical solver. This non-physical process is extremely sensitive to the mesh density at the trailing edge; it impacts: the separation point, pressure in the re-circulatory region, and the overall drag of the aerofoil. This sensitivity is visible in the relative swing of 5.53 drag counts between the optimised profile and the truncated ogive when going from a coarse optimisation mesh to a finer analysis mesh. It is interesting to note that the optimised and analytical profiles are affected differently by mesh refinement: while the separation point moves back on the optimised profile due to the better resolution around the curved trailing edge leading to a drag increase; the perfectly sharp corner of the analytical profile the same separation point even if the base pressure is changed, and the drag is reduced.

The case for an area constraint of 0.01 is an outlier, the optimiser failed to find the best solution after a change of topology close to the leading edge happened on the first line search of the optimisation. These early changes of topology are not unusual in the line search; they are usually recovered from, however due to the extreme thinness of the profile and the relatively coarse cell height of the optimisation mesh, the optimiser could not find the suitable decent direction.

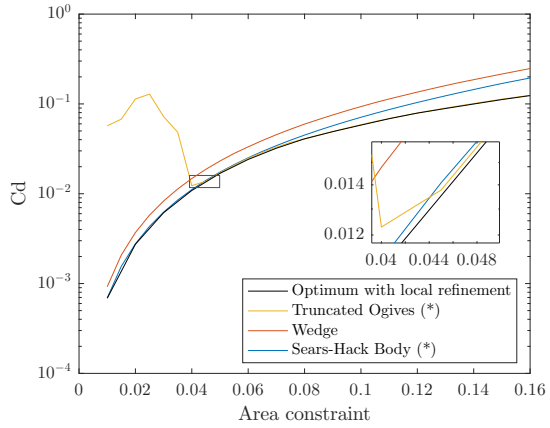
6.2.1 Impact of Refinement on Aerodynamic Optimisation Case

The same suite of cases was repeated using the refinement criterion specified in sub-Section 4.4.2. The optimisation was started from a 2 by 4 Cartesian grid of RSVS design variables with symmetry and five refinement steps were carried out. The goal of the refinement is to do away with the need for expert knowledge when setting up an optimisation method for a specific case improving the robustness of the entire process. The combination of the refinement criterion and the aerofoil like cases, naturally leads to refinement splitting cells almost exclusively in the vertical direction; a behaviour similar to refinement in aerofoil specific parameterisation.

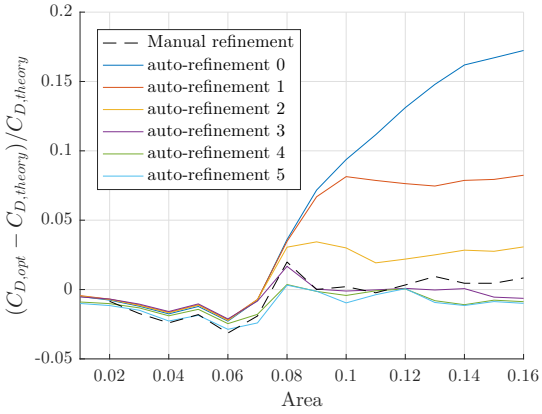
Drag results and optimum profiles for these aerodynamic cases are shown in Figure 6.5. Figure 6.5a shows the evolution of the drag coefficient for the analytical and non-linear optima with refinement. The inset shows the behaviour of the different optima between 0.04 and 0.048, importantly the optimisation framework successfully negotiates this complex region where two theoretical optima exist. Figure 6.5b shows the difference in drag value between the best analytical optima and each stage of the refinement process.

For low values of area the first refinement stages are sufficient to exceed the analytical optimum. Figure 6.5c shows the optimum profiles for each value of the volume constraint. The cases up to areas of 0.07 result in profiles close to parabolic but as the required volume increases, the point of maximum thickness is shifted towards the trailing edge. This finding is similar to a previous study by Palaniappan and Jameson [247]. This behaviour allows the shock to be weaker for the non-linear optimum than for the corresponding ogive, which more than makes up for the increase in back pressure. This simple geometric behaviour is easily captured by RSVS geometries with or without refinement.

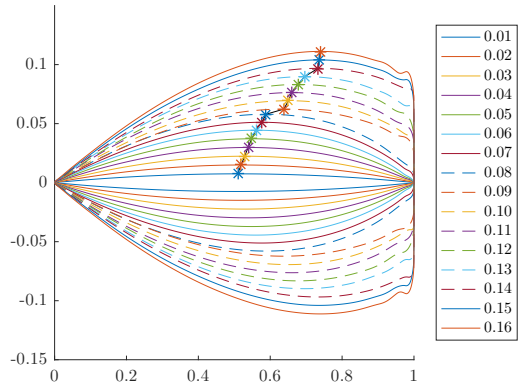
For values of area above 0.09 it is shown that the profiles tend to the truncated ogives of Klunker and Harder [246]. The large discontinuity poses a challenge to the parameterisation leading to difficulties for the optimisations. However, refinement enables the blunt trailing edge to be represented to a sufficient level and significantly improves the optimum that could be recovered compared to the cosine grid with smooth design variables. The behaviour through the refinement stages seen in Figure 6.5bis similar to that observed for the inverse design case presented earlier: each refinement stage unlocks a new portion of the design space to significantly improve the objective function. This helps to validate the use of curvature as a measure of the need for finer parameterisation. While there is no change in the number of bodies these results are enabled by the



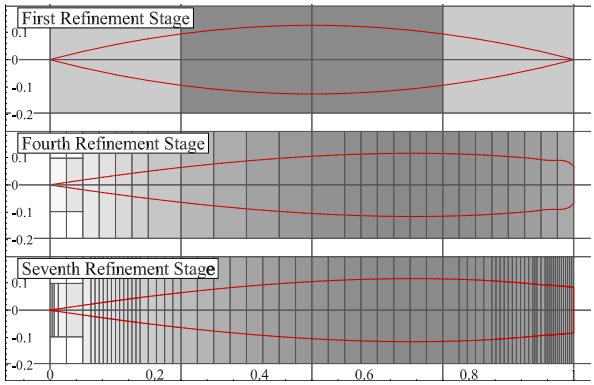
(a) C_D values for single body analyticaloptima (* starred in the legend) and non-linear optima for changing area constraint.



(b) Drag difference between analytical and non-linear optima



(c) Optimised profiles with points of maximum thickness (*)

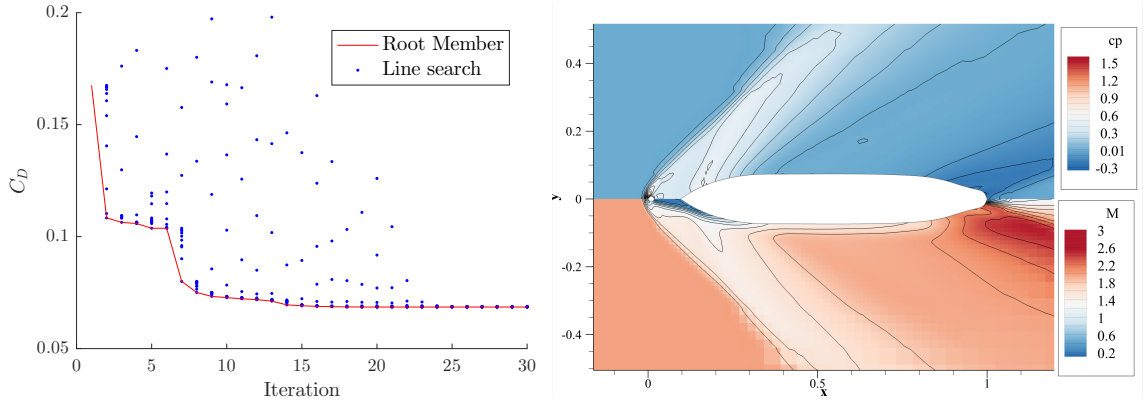


(d) RSVS grids at 3 stages of refinement for a constrained area of 0.13

Figure 6.5: Summary of Supersonic Optimisation results using local refinement and smoothing compared to the best single body analytical solutions.

flexibility of the parameterisation and flow solver, traditional parameterisations are not always capable of transitioning between smooth and sharp corners as is required by the larger area cases.

The higher area cases, notably 0.15 and 0.16 exhibit a small oscillation of the profile at the trailing edge (Figure 6.5c). These oscillations are the result of the optimiser minimising the turning circle of the flow to favour inviscid separation when the design variable resolution is insufficient to represent a blunt trailing edge. The large area of re-circulatory flow behind the blunt trailing edge leads to poor flow convergence and poor quality gradients, preventing the optimiser from recovering at higher refinement levels. This highlights a limitation of using an Eulerian flow solver for this case: physical modelling of the boundary layer and separation is required.



(a) Convergence history of the CG process

(b) Pressure coefficient and Mach flood plots for iteration 4.

Figure 6.6: Aerospike profile appearing at iteration 4 of the CG optimisation for an area constraint of 0.11.

The RSVS parameterisation shows that there are two different flow regimes that result in optimum solutions: attached flows for low areas and detached flows at the trailing edge for higher areas. This can be seen clearly in Figure 6.5c: the points of maximum thickness show clear separate trends for the parabolic profiles (up to an area of 0.08) and the truncated ogives (after an area of 0.09). These results also highlight the existence of possible multi-modality around an area of 0.08.

6.2.2 Impact of Topological Flexibility on Aerofoil Optimisation

The case with an area constraint of 0.01 showed that, while the topological flexibility is not expressly needed to solve this suite of shape optimisation, it still has an impact on the optimisation behaviour. In that case the impact was negative, however it can also reveal interesting designs within the constraints of the aerofoil design space. Figure 6.6 shows the flow around the best profile at iteration 4 of the drag minimisation of a profile of area constraint 0.11, for which the full results are available in Figure 6.3.

At the 4^{th} iteration, the CG optimiser has found a two body profile to be beneficial. This profile is reminiscent of the *aerospike* configuration used to reduce drag on the Lockheed Martin Trident D-5 submarine launched ballistic missile. The aero-spike is an actuated drag reduction device which extends out of the missile, forming an oblique shock ahead of the blunt nose of the missile; significantly increasing its range. This type of behaviour shows the capability of the topological optimisation framework as a tool for exploratory design studies.

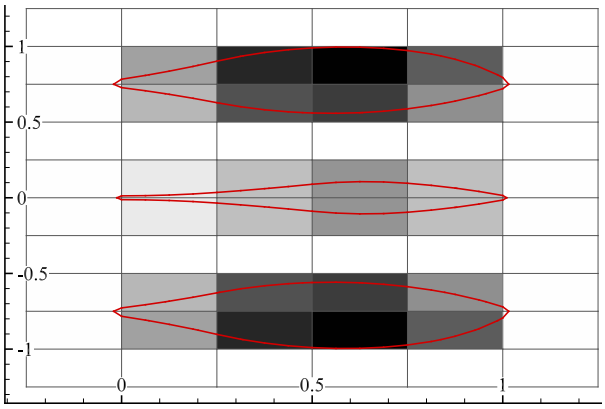
The flattening of the convergence history (Figure 6.6a) between iterations 2 and 6, close to the topology change, suggest the optimiser is converging on a local optimum. This convergence shows how topology change creates additional complexity, even in a small design space. This means that effective topology optimisation requires methods for exploring points beyond a local minimum, warranting the use of global optimisers.

6.3 RSVS Topological Aerodynamic Optimisation

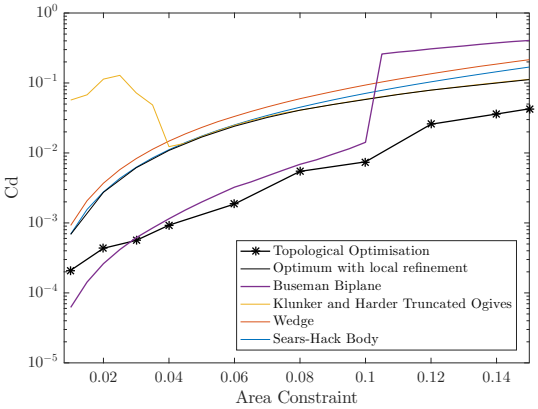
The approach used for the supersonic single body area constrained optimisation cases in the previous section is repeated for the multi-body case. In those cases the non-linear optimum is compared to the drag value of the Busemann bi-plane. The topological flexibility is enabled by using a larger RSVS grid layout, in this case a 10 by 6 VOS grid was used. Symmetry of the profile was enforced by mirroring the VOS design variable values: meaning the optimiser controls 30 effective design variables. To ensure exploration of the layout is effective the differential evolution (DE) optimiser is used. The starting population is of critical importance to ensure that exploration is sufficient but convergence is quick. The optimisation starts from a family of random multi-plane profiles with sharp trailing and leading edges. Starting from a population of bodies with good aerodynamic qualities reduces the convergence time significantly while still allowing the design space to be explored. A population of 100 was used as smaller populations showed inconsistent behaviour in repeated runs. The optimisation was stopped once the population showed no topological diversity. This was assessed through the convergence of the population on a set of non-zero VOS design variables, This occurred between iterations 200 and 300.

Figure 6.7 presents the drag results and profiles resulting from the topological optimisation process. Figure 6.7b shows the drag of the optimised profiles is below that of the Busemann biplane or the best single body analytical optima for all values of the constraint above 0.02. This good performance above 0.02 is because the optimisation tends to build very efficient profiles which resemble convergent divergent nozzles with flat outer edges. The smooth compression which results from these has a much lower drag on a discrete grid compared to the Busemann bi-plane which relies on perfect shocks and expansion fans.

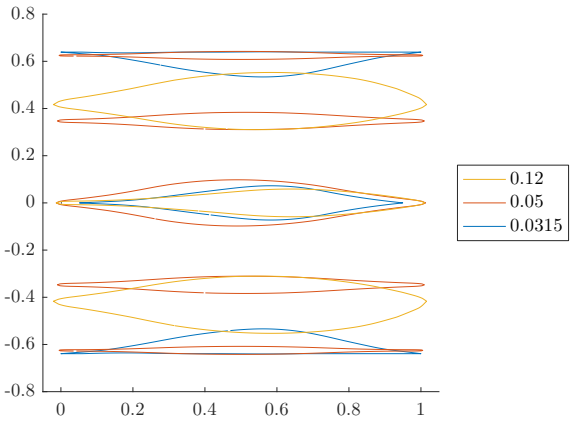
Below 0.02 the profiles are extremely thin, building planes less than 1% thick. This means that the sharpness of the leading and trailing edge play an outsize role in the quality of the optimum. Resolving and achieving a perfectly sharp leading edge is a



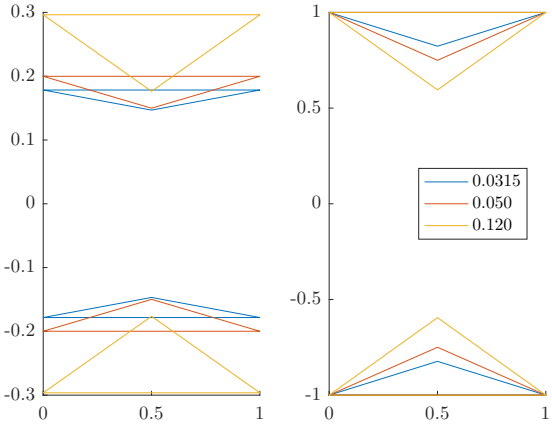
(a) RSVS Grid and optimum profile for a constraint of 0.12



(b) C_D values for analytical and topological optima for changing area constraint

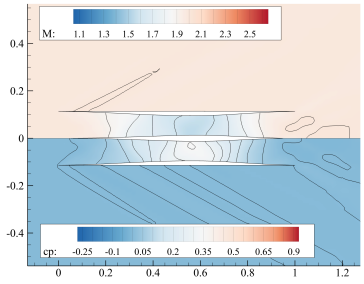


(c) Optimum profiles scaled by the height of a Busemann bi-plane of the same volume

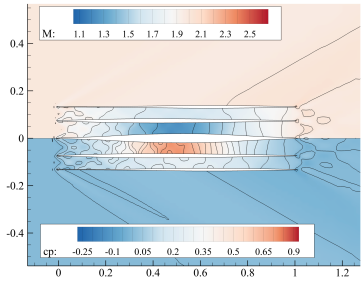


(d) Busemann bi-planes, unscaled (left) and scaled with $\Delta y_{BUSEMANN}$ (right)

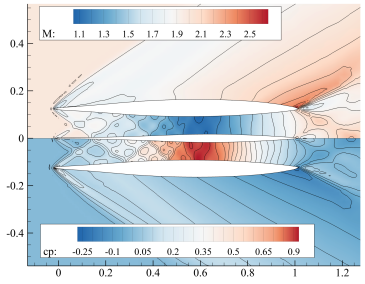
Figure 6.7: Results of Supersonic Topological Optimisation for a range of area constraints compared to analytical solutions.



(a) Optimised profile with an area of 0.0315 after 400 iterations

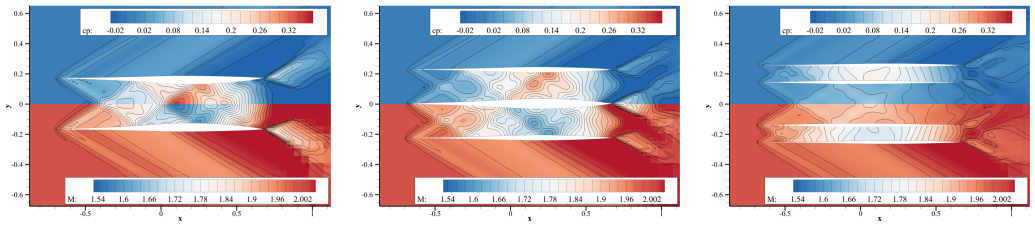


(b) Optimised profile with an area of 0.06 after 300 iterations

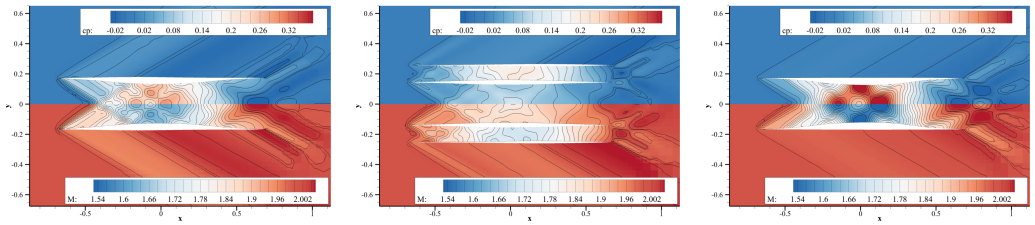


(c) Optimised profile with an area of 0.12 after 300 iterations

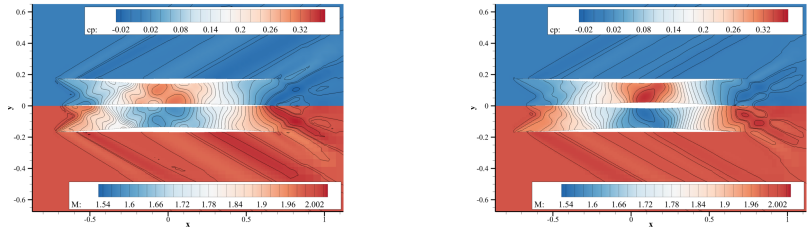
Figure 6.8: Flood plots of Cp and Mach number for Supersonic Topological Optimisation results presented in Figure 6.7c.



(a) Best profile at iteration 1 (b) Best profile at iteration 2 (c) Best profile at iteration 32



(d) Best profile at iteration 53 (e) Best profile at iteration 71 (f) Best profile at iteration 130



(g) Best profile at iteration 234 (h) Optimum profile at iteration 400

Figure 6.9: C_P and Mach flood plots for the best solutions at selected iterations

challenge to all the elements of the framework. The drag values are very small leading to a lower number of significant figures for the objective. This affects the already limited capacity of the global optimiser to converge locally. Finally perfectly sharp corners exist at the limit of what the parameterisation can represent, making them difficult features to achieve with a global optimiser which tends to progress in slightly erratic steps.

Above an area of 0.1 the flow in the Busemann bi-plane is choked which can be seen in the very large increase in drag (18 times larger between areas of 0.1 and 0.11). In Figure 6.7c the profile of area 0.12 displays the internal features similar to the optima of area 0.0315 but with curved outer edges. Figure 6.8c shows that the optimiser is combining the flow features of an optimised Busemann bi-plane (flow similar to Figure 6.8a) with one of the single body optima of Figure 6.5 (flow similar to 6.1c). This allows low drag to be maintained where a traditional Busemann bi-plane would choke.

The optimum profile for an area of 0.06 shows a penta-plane profile (Figure 6.8b),

while for 0.0315 the optimum is a tri-plane (Figure 6.8a). The main factor in drag reduction is the minimisation of external shocks; because these shocks can be removed with any multi-body profile, different optimum topologies for different runs can be expected. The optimality of different topologies also explains the large number of iterations required for topology to converge in Figure 6.9: the optimum topology is not stable before the 234^{th} iteration. This behaviour of the optimisation problem is a clear sign of multi-modality in the formulation of this problem, with one local optimum solution for each profile topology. It also appears to be a sign of degeneracy: there are many profiles which achieve very similar, locally minimal objective function values, with different topology. The degeneracy arises as it seems probable that for any given area there may be infinite multi-plane profiles solving the drag minimisation.

The combination of parameterisation, global optimiser and flow solver is effective in exploring this problem. The relative compactness of the set of design variables as well as the smoothness of the recovered profiles ensures that good aerodynamic bodies are generated most of the time without arbitrarily restricting the design space. This allows this topological optimisation framework to explore the complex behaviour of the optimal solution for large values of area.

These differences in optima highlight a limitation of the differential evolution on this topological aerodynamic optimisation case. While DE provides good exploration, the convergence on the global optimum is not guaranteed; this is because each of the local minima has a very similar drag value but with different topology. Alternate algorithms for niching and hybrid gradient/agent search methods could help improve the performance of the framework on these cases. Despite these limitations, the combination of parameterisation, global optimiser and flow solver is effective at exploring these optimisation problems. The relative compactness of the set of design variables as well as the smoothness of the recovered profiles ensures that good aerodynamic bodies are generated most of the time without arbitrarily restricting the design space. This allows this topological optimisation framework to explore the complex behaviour of the optimal solution for large values of area.

6.4 Integrated Optimisation Framework

Previous sections have shown that an ASO framework with the RSVS parameterisation is an effective tool to tackle a range of optimisation problems. It was capable of exploring shape optimisation problems as well as a family of topology optimisation problems,

highlighting interesting aerodynamic behaviours and optima. However some limitations were uncovered notably in terms of local convergence and handling of degeneracy in the optima.

This[2] section brings together the RSVS framework with the MLSO developed by Masters et al. [176] which was shown to be the most effective ASO framework on benchmark problems. The integration of the flexibility of the RSVS and the efficiency of the MLSO lets each parameterisation framework tackle the part of the problem it is most suited to, allowing efficient and thorough exploration of the aerodynamic topological design space.

The work on "integrated parameterisation" is the result of a fruitful collaboration with Kedward; the concern of this thesis remains in the implications of aerodynamic results, especially regarding the modality of the optima, and the validation of the method for tackling a wide range of problems. The implementation of the MLSO was the work of Kedward, as a result only brief summaries of the methods are presented in the main body of the thesis, please refer to Appendix D, Kedward et al. [248] and Payot et al. [249] for full implementation details.

6.4.1 Integrated Geometry Parameterisation

The integration of the RSVS with the MLSO framework into a single geometry generation framework aims to enable very efficient aerodynamic exploration of topological design spaces. It brings together efficiency and topological flexibility to reduce computation cost on already explored cases, and to enable increased design space exploration for new cases.

This section details how two sets of design variables can be handled concurrently in a unified optimisation process. Because the multi-resolution subdivision formulation allows any input geometry to be the starting point of the multi-resolution framework (see Figure 6.10), the RSVS can be used to introduce topological flexibility. In this integrated framework, the design geometry is controlled by both the RSVS design variables (A_j of Equation 3.1) and a coarse set of $\mathbf{p}_k$ control points with seamless transfer of information between the two.

[2]This work was performed in collaboration with Kedward and form the basis of conference publications [248, 249]. Any aspect of the work on which the author of this thesis did not make a significant contribution, namely the MLSO framework, is summarised in the main body and is presented in detail in the appendices.

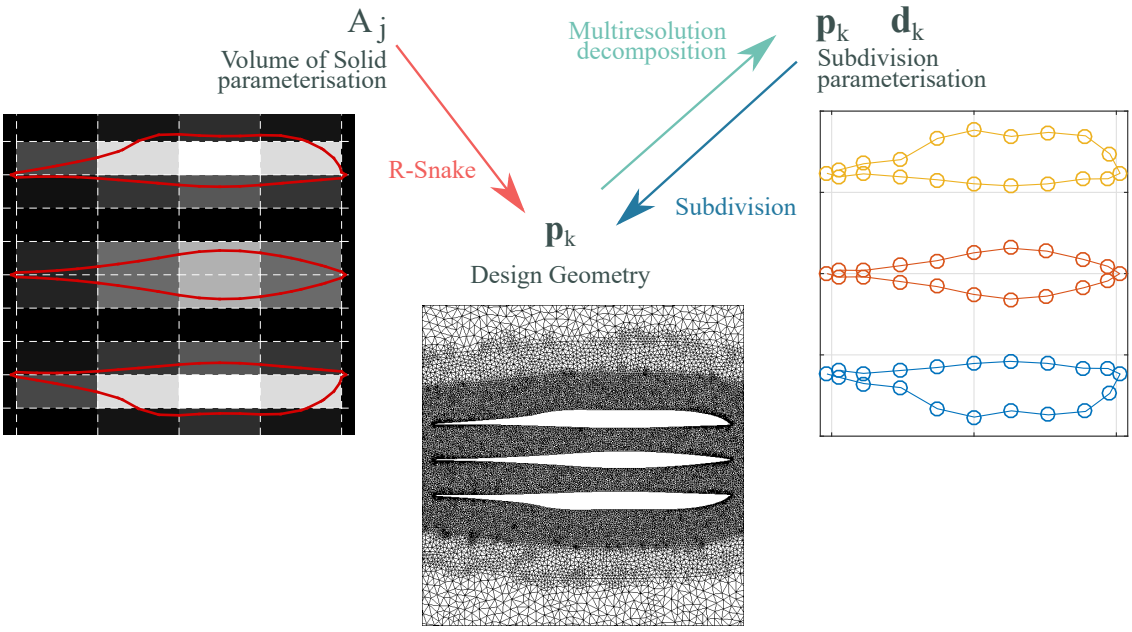


Figure 6.10: Transfer of geometry between parameterisations from [248].

Algorithm 6 Integrated Parameterisation Optimisation Framework (from [248]).

for Each global optimisation step **do**
Generate RSVS Profiles
Re-parameterise using multi-level subdivision curves
Perform local (topology invariant) optimisation
Translate subdivision deformations onto RSVS design space
Generate new population of RSVS design variables using a global optimisation process
end for

The next section (Section 6.4.1.1) highlights the main features of the MLSO parameterisation approach for a full description of the method the reader is referred to Appendix D and the publication of Masters et al. [176]. The integration of the two parameterisations is presented in Section 6.4.1.2, showing how the two sets of design variables can be controlled sequentially or simultaneously to achieve topological flexibility with a compact design space (see Algorithm 6).

6.4.1.1 Efficient shape optimisation using multi-resolution subdivision

A brief overview of the key features of the MLSO parameterisation method as developed by Kedward are provided here, for a full account of the derivation of the method please

refer to Appendix D. The same properties that make subdivision curves an efficient smoothing operator for geometry processing make it a very capable parameterisation method. Subdivisions were discussed in the context of the control of the surface distribution of RSVS profiles in Section 5.2.1. Subdivision curves define a refinement operator which sequentially defines a finer set of points ($\mathbf{p}_{k+1}$) as the linear combination of an initial coarse set of control points ($\mathbf{p}_k$). The refinement operation defined by the matrix $\mathbf{S}_k$ was initially specified in Equation 5.4 repeated below.

$$\mathbf{p}_{k+1} = \mathbf{S}_k \mathbf{p}_k$$

The subdivision formulation can be used for parameterisation because the limit as refinement is performed infinitely many times can be exact splines for a specific layout of parameters: in work B-splines equivalent subdivision is used. This means that points $\mathbf{p}_k$ are the control points of a spline $\mathbf{R}_s$ through the basis matrix ϕ_k. In order to recover arbitrary shapes from a given subdivision level detail vectors $\boldsymbol{\delta}_k$ can be added to the formulation. Unlike previous subdivision based parameterisation methods, the linear parameterisation used in the integrated framework is augmented by adding control over the intensity of the detail vectors through additional design variables β^x and β^y defined by Equation 6.3.

$$\begin{pmatrix} \mathbf{p}_f^x \\ \mathbf{p}_f^y \end{pmatrix} = \begin{pmatrix} \phi_k & \mathbf{0} & \mathbf{e}_k^x & \mathbf{0} \\ \mathbf{0} & \phi_k & \mathbf{0} & \mathbf{e}_k^y \end{pmatrix} \begin{pmatrix} \mathbf{p}_k^x \\ \mathbf{p}_k^y \\ \beta^x \\ \beta^y \end{pmatrix} \qquad (6.3)$$

$$\text{with :} \qquad \begin{pmatrix} \mathbf{e}_k^x & \mathbf{e}_k^y \end{pmatrix} = \sum_{i=k}^{N} \phi_{i+1} \boldsymbol{\delta}_i$$

This multi-level representation is an extremely powerful tool for parameterisation as it allows any profile to be exactly represented at many control levels. In this way, coarse and compact design variables are provided by control points $\mathbf{p}_1$ which allow broad changes; while fine geometric modifications can be performed with the higher refinement levels ($\mathbf{p}_N$). The augmentation of the basis was found to be more robust and effective than ignoring the error vectors or keeping them constant, respectively.

6.4.1.2 Parameterisation interface

Transfer of shape information from the RSVS contours to the subdivision parameterisation is done via high fidelity discretisation of the body profiles. Each RSVS case may

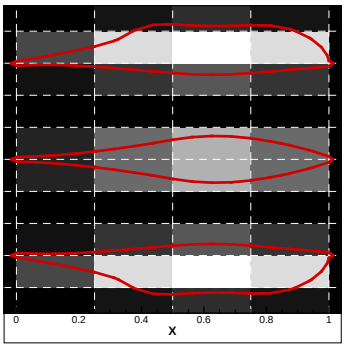
contain multiple bodies, each of which is re-sampled to be represented by a piecewise linear *'loop'*. The set of loops provides a general representation of the geometry which allows modular interfacing with the local shape parameterisation method.

While any input surface can be parsed by the combined framework, surface point distribution has an impact on the quality of the MLSO subdivision basis. A smooth surface distribution significantly improves optimiser behaviour; this is achieved by re-sampling the RSVS contours requires with smooth, symmetric clustering of points around sharp corners. The generalised cosine re-sampling process outlined in Section 5.2.2 was used to ensure seamless integration. Unlike traditional aerofoil optimisation cases there is no a priori knowledge of what the geometry might look like and where clustering might be required. Sharp features requiring clustering of surface points are identified using the exterior angle at each point, smoothed using a moving average over 3% of the points. This measure allows sharp corners (similar to trailing edges) as well as areas of sustained curvature (similar to leading edges) to be identified as extrema of the function. The region of the profile around each identified feature is then sampled using a cosine distribution parametrised by edge length. This process allows symmetric point distributions around features and regions of consistently low curvature have a low surface resolution reducing the computational fluid dynamics (CFD) mesh density and computational time.

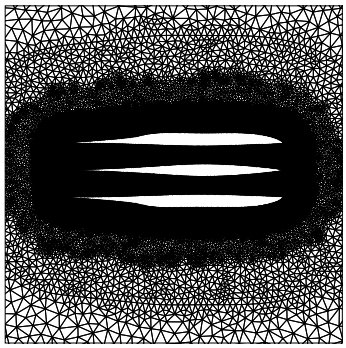
The geometry is then re-parameterised using subdivision curves for use in the MLSO process. The re-parameterisation is as follows:

- define a high fidelity cubic B-spline for each loop, this defines $\mathbf{R}_s$ without the need for infinite subdivision;
- every loop is split into continuous regions, with a spline fit by least squares, this enables the preservation of sharp points;
- this spline is reverse subdivided to form the coarser design levels;
- design levels are merged across loops to form each geometry design level.

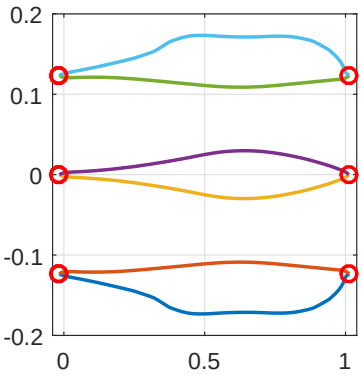
The integration process is illustrated for a three body case in Figure 6.11. The effectiveness of this approach is first shown on the geometric shape matching of an aerofoil with the flap in Figures 6.12 and 6.13. This shape matching case minimises the "area error" measurement defined in Section 5.1.2. It illustrates the MLSO framework recovering exactly an RSVS profile and exploiting five subdivision levels to arrive at an accurate representation of the profile. As expected, the MLSO framework is extremely effective on the fixed topology problem, outperforming the shape optimisation capability of the RSVS with refinement. These initial results serve to validate the effectiveness



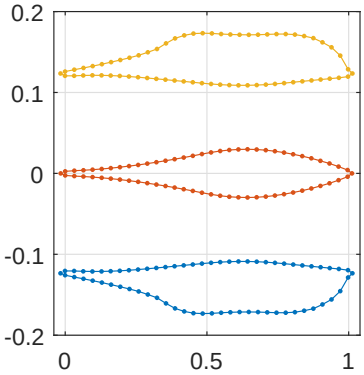
(a) VOS grid with r-snake profile (in red).



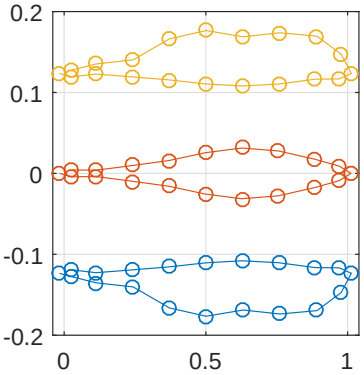
(b) Discretised surface and resulting triangular mesh).



(c) Identify continuous regions between sharp points



(d) Create base spline



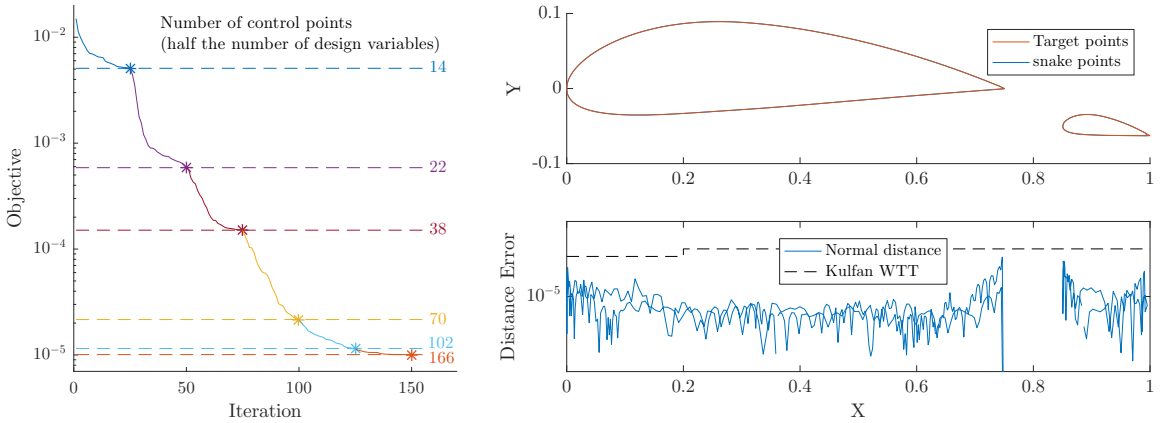
(e) Reverse subdivide

Figure 6.11: Automated generation of subdivision parameterisation (from [248]).

of the combined framework and demonstrates its potential to outperform the RSVS on aerodynamic topology problems.

One additional challenge in the effective implementation of the combined MLSO-RSVS framework is the formulation of appropriate constraints on the geometry. Without a priori knowledge of the topology and the geometry of a profile many of the traditional ASO constraints which prevent pathological behaviour need to be reformulated. The constraints devised by Kedward applied to each geometry are the following:

- **Overall maximum chord:** the chord-wise extents of all bodies is bounded;
- **Outer move limit:** shape displacements in the outward normal direction must be within a polygon defined around each loop by offsetting the loop convex hull;
- **Inner move limit:** shape displacements in the inward normal direction must not cross the initial camber line of the loop.
- **Surface element size:** surface elements cannot shrink or grow beyond specified



(a) Convergence history of the optimisation process

(b) Profile distance between the multi body aerofoil and the final profile

Figure 6.12: Convergence history and final profile error for the Geometric Inverse design of the multi-body aerofoil using the MLSO framework starting from an RSVS geometry.

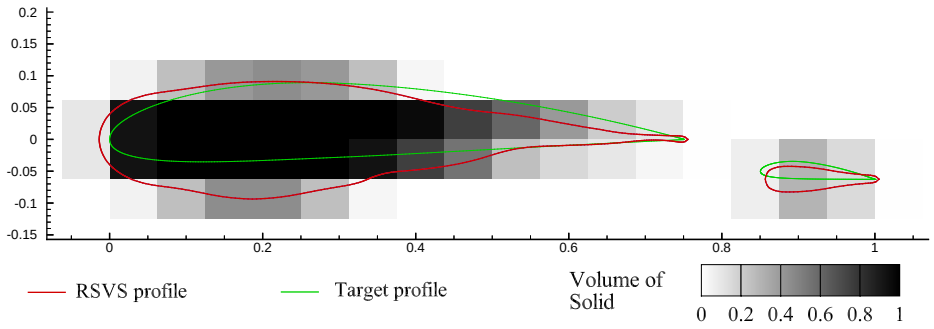
fractions of their initial size.

In essence the "move limit" constraints are needed to restrict the MLSO to a single topological state of the RSVS: beyond those limits topology would need to change which is not possible in the MLSO framework alone.

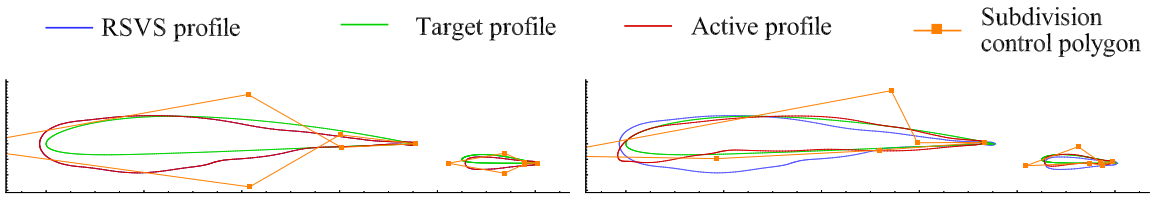
While formulating such constraints is possible for specific problems with knowledge of the geometries likely to be optimal; it is very difficult to make them general enough in a way that enables sufficient design space exploration but avoids those non-physical designs. These limitations highlight the benefit of a parameterisation method, like the RSVS, capable of natively handling all these constraints implicitly.

6.4.2 Modifications to the Optimisation Framework

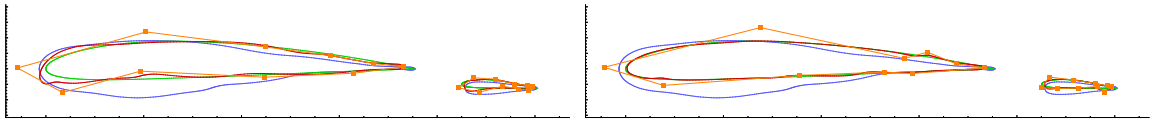
To support the integrated MLSO-RSVS parameterisation the optimisation framework presented in Section 5.1 needed to be enriched with new optimisers, flow solvers and meshers; which support the evaluation of adjoint sensitivities. Section 6.4.2.1 presents the global and local optimisation methods, followed by a description of the hybrid scheme used to combine global and local search methods. The increased complexity of this hybrid approach arises due to the unsupervised initialisation and running of gradient-based sub-optimisation problems. The implication of this is that reliable evaluation of the flow and sensitivities is essential; the tools and techniques used to achieve this are presented in Sections 6.4.2.2 and 6.4.3.



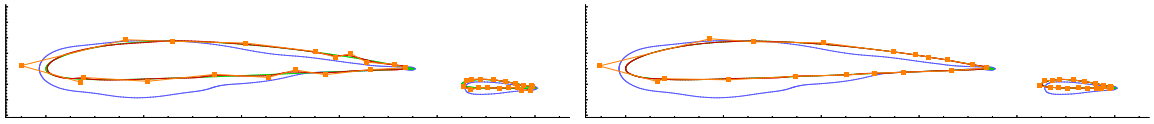
(a) Topology matching using the RSVS



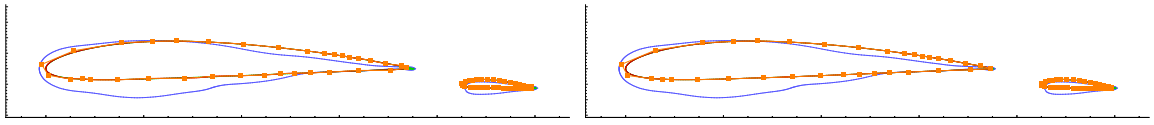
(b) Starting refinement level (level 3), matching the RSVS (left) and final profile (right)



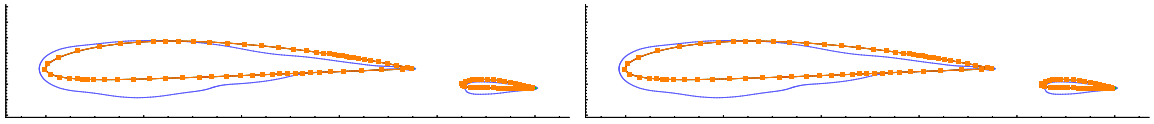
(c) Subdivision level 4



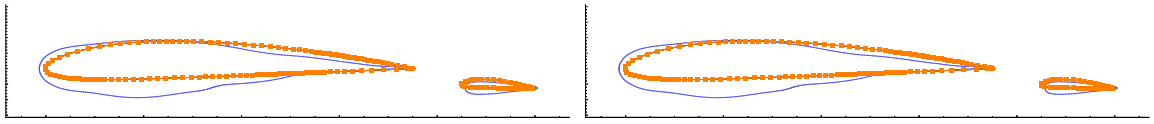
(d) Subdivision level 5



(e) Subdivision level 6



(f) Subdivision level 7



(g) Subdivision level 8

Figure 6.13: Geometric matching of a multi-body aerofoil using the RSVS-MLSO framework, profile represented are the start (left) and end (right) of each refinement step.

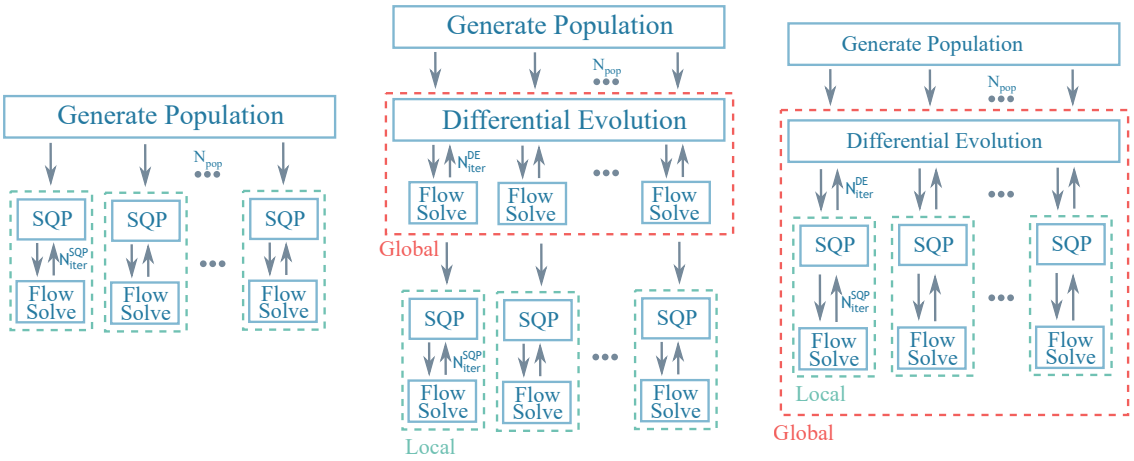
6.4.2.1 Optimisation methods

In addition to the subdivision parameterisation the MLSO framework relies on the integration with a sequential quadratic programming (SQP) optimiser and a flow solver which supports adjoint design sensitivities. SNOPT an implementation of SQP is used for shape optimisation with the MLSO; and DE (Section 5.1.1.3) exploits the topological flexibility of the RSVS. These two methods are integrated into hybrid optimisers that could efficiently and reliably exploit the integrated parameterisation.

Local optimisation using gradient based optimiser The Sparse Non-linear OPTimizer (SNOPT) [252] package is used here for gradient-based optimisation. This package implements a SQP algorithm for solving general non-linear constrained optimisation problems. The power of this package lies in its ability to efficiently and robustly handle large problems ($\approx$ 1000s of variables and constraints) while allowing precise constraint satisfaction. The SQP algorithm operates iteratively whereby successive search directions are found from the solution of a quadratic programming (QP) sub-problem and a line-search is used to determine step length. The sub-problems are formed from quadratic approximations to the augmented objective function (Lagrangian) and linearisations of the constraints. The quadratic approximation is initialised with an identity matrix and BFGS updates are used to approach the Hessian of the Lagrangian.

The SQP gradient-based algorithm used in combination with multi-resolution subdivision curves for shape parameterisation and an adjoint flow solver (see sec. 6.4.2.2) for objective sensitivities, results in an efficient and effective tool for multi-level subdivision optimisation (MLSO). The MLSO performs sequential shape optimisations starting from a low fidelity subdivision curve, consisting of few control points, and progressively increasing. After each intermediate optimisation level there is exact transfer of the optimum result from the previous level to the starting geometry for the next level.

Hybrid optimisers In most cases of aerodynamic optimisation efficient approaches have either relied on optimisation algorithms whose convergence is linearly correlated with number of design variables [253] or used very low number of design variables with global optimisers [79]. Neither of these approaches is satisfactory for the cases to be tackled by the integrated parameterisation method; the topological flexibility leads to a multi-modal design space which cannot be sufficiently explored using local optimisers and requires too many design variables for the routine use of a global optimiser.



(a) Multistart local shape optimisation

(b) Sequential 'global then local'

(c) Hybrid algorithm

Figure 6.14: Algorithms for combined global topology and local shape optimisation (from [248])

To tackle these shortcomings three procedures were considered: multi-start (MS) local, sequential global to local (SGL), and a hybrid optimisation approach. The multi-start local optimisation uses a randomly generated sample of profiles and optimises all of them, extracting the best result. The sequential approach relies on a set number of DE steps, then uses this population to begin a MS gradient-based optimisation method. The hybrid optimiser is similar to that used by Chernukhin and Zingg [58]. It relies on performing a few local optimisation steps on each member of a global optimisation population before calculating new generations. This approach was shown to be effective in cases where a high number of local minima are present [58]. These three approaches are illustrated in Figure 6.14. Here the gradient-based shape optimisation, performed by MLSO, is identified as the *local* search, and the topological optimisation, performed using RSVS with DE, is identified as *global*. For the sequential implementation, the interaction between the two is only weak whereas for the hybrid method they are more strongly coupled.

Tests using the hybrid optimiser have shown it to be ineffective in the current multi-parameterisation set-up. This is due to the poor "aerodynamic potential" of many of the profiles generated by DE when using the RSVS. For these profiles a simple flow solution is sufficient: any additional computational effort is wasted. For this reason, a small modification to the hybrid optimiser is proposed: instead of performing MLSO on every DE step, the global optimiser will be allowed to perform more than one DE

step before calling on the MLSO framework for accelerated convergence. This two step process exploits the ability of DE to discard the poorest profiles, while maintaining a population which will benefit from a subsequent SQP exploration.

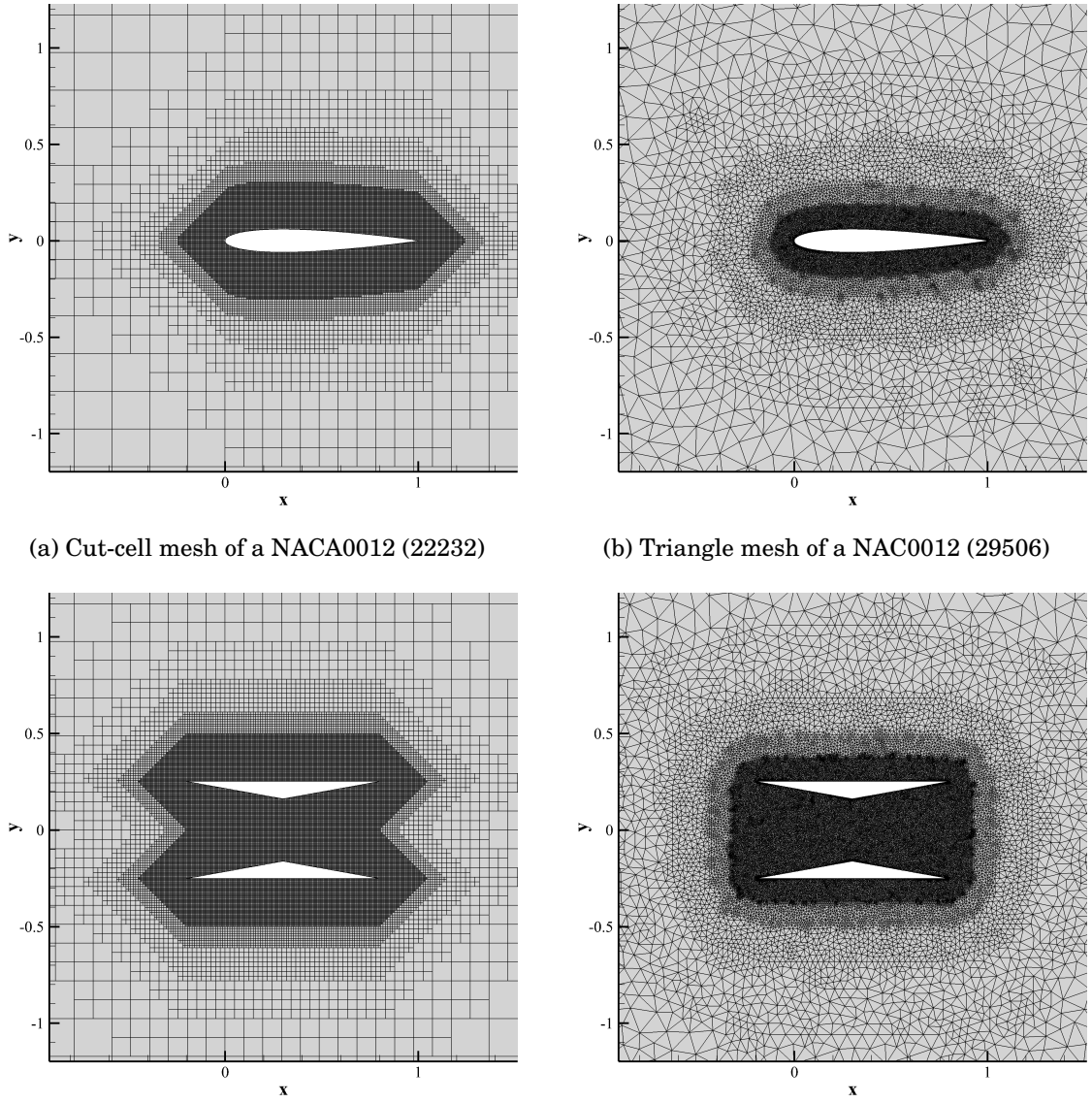
6.4.2.2 Flow analysis and discretisation

The global search uses the RSVS only framework described in Chapter 5 and used in Section 6.3. It uses a cut-cell mesh generator for topological flexibility with at low computational cost [152, 153] with an unstructured cell-centred flow solver [239, 240]. The cut-cell meshes were used in conjunction with the DE optimiser.

the Stanford University Unstructured (SU2) [254] is used in conjunction with the MLSO method. SU2 provides Euler and RANS flow unstructured finite volume flow solving along with continuous and discrete adjoint for the evaluation of flow sensitivities [255]. The MLSO uses the Euler flow solver, the continuous adjoint, multi-grid and the MPI parallel option.

The cut-cell meshes used in the RSVS only framework are not compatible with SU2, and had to be replaced with triangular meshes. Triangular mesh generation is performed in an automated manner using 'Triangle' implemented by Shewchuk, a robust, light-weight mesher using constrained Delaunay Triangulation [256]. This method allows bounds to be placed on internal mesh angles, providing a cheap and robust method for generating volume grids of acceptable quality for the large variety of shapes and topologies encountered in the topology optimisation process. Element size is constrained to mimic the element size behaviour of cut-cell meshes: as distance from the profile increases so does the element size. Projected convex hulls of the profile geometry are used to define regions of uniform mesh density in which the size constraint is relaxed as the region is further from the geometry. Figure 6.15 shows close-ups of the resulting triangular meshes and cut-cell equivalents with the same cell height at the surface. Unsurprisingly, the triangular meshes require a higher element count for a given cell height. This increased cell count comes at a slight computational penalty but comes with the ability to cluster cells in areas of high curvature which allows improved shock resolution, a key component of accurate drag prediction in supersonic flow.

Table 6.3 presents the solver settings for SU2. These were devised by checking the convergence of the flow solver and the adjoint solver on 100 profiles generated by the RSVS and meshed by Triangle. These settings maximised the number of converging flow and adjoint solutions. The dissipation on the adjoint flow solution is critical to ensuring reliable convergence on the triangular meshes used in this work.



(a) Cut-cell mesh of a NACA0012 (22232)

(b) Triangle mesh of a NAC0012 (29506)

(c) Cut-cell mesh of a Busemann biplane (37828)

(d) Triangle mesh of a Busemann biplane (59564)

Figure 6.15: Triangular and cut-cell meshes for similar edge lengths constraints (number of elements in parenthesis).

During shape optimisation, mesh deformation is used to produce new meshes for the displaced surface geometry from the initial volume mesh. Not only is this computationally cheaper than regenerating a mesh for each geometry iteration but it also maintains consistency of the discretisation error which is highly desirable during iterative numerical optimisation. Interpolation with multi-scale radial basis functions (RBFs) [243] is used for mesh motion. Interpolation using radial basis function (RBF) has recently become a prominent mesh deformation method boasting excellent robustness and

Table 6.3: SU2 Configuration

Physical problem	Compressible Euler	Continuous adjoint (C_D)
Convective method	Jameson-Schmidt-Turkel	
Time integration	Euler implicit	Runge-Kutta explicit
Artificial Dissipation (k_2, k_4)	$0.75, 0.03$	$2.0, 0.08$
Target Residual	10^{-10}	10^{-10}

quality-preserving characteristics [146, 257, 258]. The multi-scale formulation varies the length scale of the interpolant such that the system solution and update steps are cheap and well-conditioned, while still recovering the exact surface displacements [243].

6.4.3 Automated Optimisation

The MLSO process must be completely automated and extremely robust in order to tolerate successfully integrate with agent based RSVS optimisation. For the parameterisation this automation and robustness relies on the integration process outlined in Section 6.4.1. For the SNOPT/SU2 local shape optimisation, all failures are captured and appropriate fall-back logic is implemented for each case. To maximise the effectiveness of the hybrid optimisation process the following fall-back routines can be introduced:

- **Flow solver contingency:** the line-search of SNOPT provides some intrinsic robustness to a failed flow solver attempt.

- **Adjoint contingency:** when the adjoint does not converge, routines automatically adjust time-stepping method, CFL number and artificial dissipation, in a predefined way, attempting to converge.

- **Re-meshing:** When the mesh deformations become high (defined by a preset geometric step), or the flow solver or adjoint repeatedly fail; re-meshing of the geometry is triggered. The re-meshing conserves the surface mesh and the MLSO parameterisation.

- **Initial objective:** in case of an unrecoverable failure during local optimisation, the initial flow solver value is returned to the global optimiser; providing some meaningful information about the profile.

6.4.4 Configuration and Validation of the MLSO

The purpose of the hybrid optimisation framework is to improve the exploration of the topological design space allowed by the RSVS parameterisation. Definition of this hybrid framework necessarily includes a number of hyper-parameters controlling: parameterisation configuration, number of iterations (at each MLSO step) and population management (amongst others). The key challenge is to choose a configuration which leads to reliable and efficient execution of the local gradient-based optimisation regardless of input geometry and without human supervision. These settings must also limit computational cost and wall time to enable the exploration of families of cases. As this mostly concerns the configuration of the MLSO only summaries of the key findings are presented below.

Selection of a robust configuration involved the repeated testing of the combined framework with different settings on the same sample of starting geometries. During testing, focus was given to both the effectiveness of the method, to build confidence that the best solution found was close to the global optimum, as well as efficiency. This validation was performed using MS gradient-based on the starting population of a run of DE for an area constraint value of $c_A = 0.12$.

The following areas were studied in the first collaborative publication [248] (also in Appendix D.4); the following findings were used to configure the MLSO:

1. **Flow solver robustness:** *does the flow solver reliably provide drag and adjoint results for the starting geometries?*
   Systematic divergence of SU2 on valid geometries initially affected the results; parameters were benchmarked on a set of geometries generated via the RSVS which was deemed representative of DE populations. Parameters presented in Table 6.3 are the result of systematic tuning for the highest convergence rate of the flow solver. Acceptable convergence rates were set using the in-house unstructured edge based flow solver used in earlier sections.

2. **Usage of the subdivision error basis:** *How should the error treatment presented in Section 6.4.1.1 be used in optimisation?*
   Exclusion, constant and variable subdivision error bases were tested in the MLSO framework. Optimisation results including the 'error basis' as design variable consistently outperformed the other options in terms of final result and reliability.

3. **Effective local optimisation:** *in what order should subdivision levels be used to*

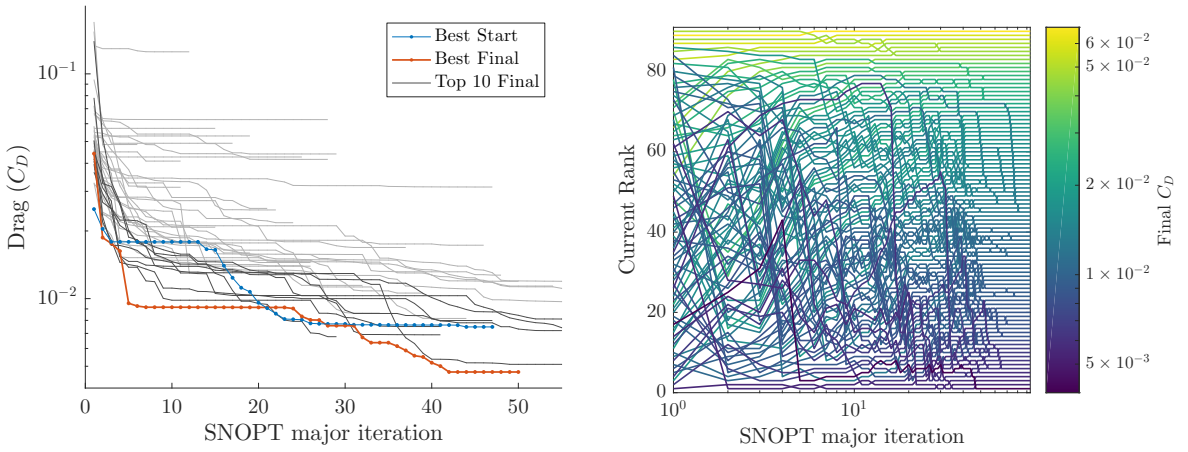
ensure reliable descent?
Starting from the coarsest available subdivision level was beneficial, it led to the largest and most reliable objective function improvement in the test population. It also allowed the largest geometric step, indicating wider design space exploration. If a single subdivision level were to be used, the third level was found to be the most effective.

4. **Efficient local optimisation:** *what is a acceptable number of local iterations to perform at each multilevel optimisation? and in what order should subdivision levels be used?*
Systematic validation runs were used to find that ten major iterations provided a good trade-off between computation cost and minimisation: more than 70% of agents reached at least 90% of their 'potential'. A much larger number of iterations is specified for the final level. Relying on optimality was found to be too costly in this context as absolute convergence of intermediate levels is of little interest, and would actually lead to over-use of the error basis (a very poor quality mode) by the optimiser.

6.4.5 Interactions with Differential Evolution

Focussing now on the global search performed by DE, two possibilities arise for trading-off computational cost with optimisation effectiveness. First, running shape optimisation on only the better performing individuals of DE population is considered. This makes sense in that effort should only be expended on profiles which stand a chance of becoming the global optimum. However this is based on the assumption that better profiles also have equal capability to improve. Figure 6.16 shows the absolute and relative performance of each agent through the convergence of the local search; it is clear that this is not the case.

Figure 6.16a shows the local optimisation convergence for the test population using MLSO where the best profiles at the start and end of the optimisation have been highlighted. Whereas the best initial profile makes approximately a 65% improvement, the best final profile is markedly better improving its original objective value by 87% and the best starting objective by 75%. This behaviour is happening throughout the population with 3 of the top 10 final profiles coming from the worst performing half of the population; *i.e.* initial objective is not a good indication of final objective after local



(a) Individual convergence of multilevel validation runs.

(b) Agents ranked by value of Drag at each iteration (logarithmic x-axis).

Figure 6.16: Convergence and rank of each agent in a multi-start MLSO validation run, the best performing profiles are seen to come from throughout the starting population.

shape optimisation. The objective improvements shown in Figure 6.16 also illustrate the significant expansion to the design space made by the multilevel subdivision curves.

The second consideration to make for the sequential “global to local" search (fig 6.14) is the number of DE iterations required before starting local shape optimisation. DE iterations allow good global exploration and a higher quality population with fewer nonsensical solutions. However, too many iterations leads to a lack of diversity in the population because of convergence of the differential evolution process.

As previously demonstrated, local shape optimisation introduces the capability to significantly improve upon otherwise poor performing agents in the global search. The solution of the global search using RSVS and DE does not usually correspond to a minima in the high-resolution design space allowed by the combined optimisation framework. Starting the local shape optimisation from converged DE results may diminish the effectiveness of the integrated scheme by repeatedly exploring the region around the minima produced by DE. Figure 6.17 shows the global convergence of the DE topological optimisation where the results of performing multi-start local shape optimisation have been included at different starting populations. It is evident that the multi-resolution subdivision parameterisation is able to expand the design space and significantly improve upon the objective. As expected, the local shape optimisation achieves improved results when it is started from a more evolved population; especially in terms of the median result of the locally optimised population. The best result is also improved except for the last case starting after 150 DE iterations; this is partially attributable to the aforementioned

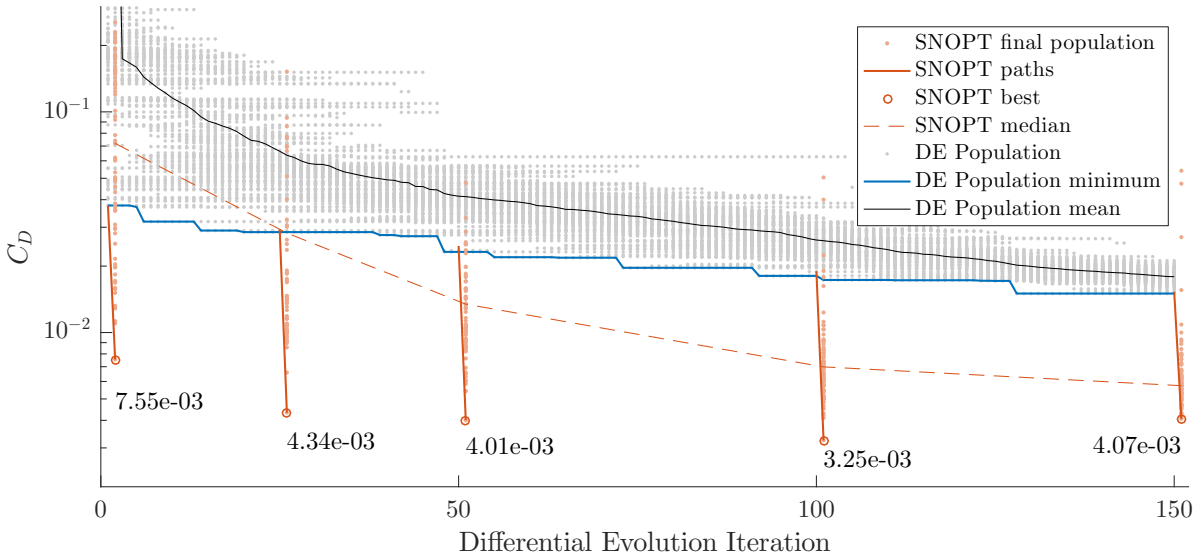


Figure 6.17: Convergence of hybrid MLSO-RSVS runs to optimise all agents from five starting points in the global search.

lack of diversity in converged DE populations.

The current population size of 100 for DE was selected to guarantee robust convergence of the DE algorithm itself. However, integration with the MLSO subtly changes the goal of the DE algorithm from an optimiser, to an agent generation method. Recognising this shift in purpose, internal DE parameters can be tuned to achieve design space exploration without the need reliable convergence on an optima. This lets the population size be shrunk reducing significantly the computational cost with only a minor penalty on the final objective value.

6.5 Optimisation Results Using the RSVS-MLSO Framework

The following section presents the results of combined topology and shape optimisations performed by the integrated framework. Previous aerodynamic topology optimisation frameworks have been hampered by their reliance on agent based optimisation. This has restricted the complexity of the cases that could be explored with a reasonable amount of computational expense. The cases tackled are supersonic drag minimisations under an area constraint for which the benefits of topology optimisation have been demonstrated by Hall et al. [155] and which were tackled with the RSVS and discussed in Section 6.3.

One of the main challenges of the MLSO-RSVS is the automation of the optimisation:

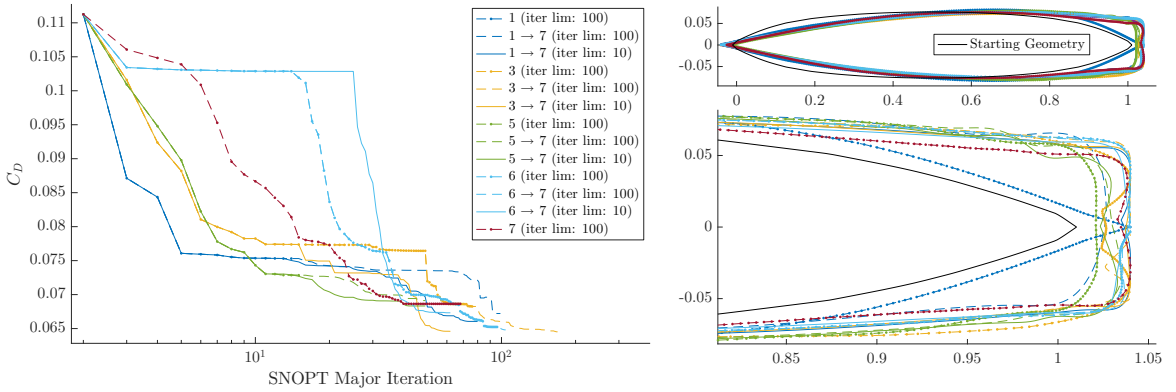
to successfully explore the design space a large number of local optimisations need to be started from a wide range of starting geometries. The process of tuning the combined method for efficiency and reliability without biasing it to specific cases has been discussed and is presented in detail in Appendix 6.4.4. First, the aerodynamic progression of the optimisers is studied in detail for a few selected cases (Section 6.5.1) showing that the desired aerodynamic features are explored. Unexpectedly, the MLSO revealed multi-modality in the shape optimisation of multi-body profiles caused by discontinuous changes in flow patterns at the optima. Finally, the results of the combined framework are shown to be superior to the previous optimisations performed using only the RSVS or the MLSO.

6.5.1 Validation of the Aerodynamic Behaviour

To validate the behaviour of MLSO on the starting geometries generated by the RSVS, a single body geometry and a two body geometry are extracted from the start population and studied in more detail. For the single body profile, the locally optimum shape is known (see Section 6.1) and can therefore benchmark the current implementation of the local optimisation framework. Similarly, study of the multi-body case aims to answer the question of modality for geometries similar to Busemann bi-planes. Each of the two cases were tested starting at subdivision levels 1, 3, 5 and 6 progressively refining up to level 7. Each level was allotted either 10 or 100 SNOPT iterations to test the effectiveness of the refinement trigger. Both cases were tested at an area constraint value of 0.12 (Equation 6.2).

6.5.1.1 Single body profile

The optimum single body profile, constrained to have an area of 0.12, is expected to resemble the truncated ogive developed by Klunker and Harder [246], shown in Figure 6.1b. This is because the reduction in the angle of the leading edge shock more than makes up for the drag generated by the back pressure applied to the trailing edge. This case presents a significant geometric challenge: the control points and the mesh need to go through an 80 deg turn to capture the flat trailing edge. Figure 6.18b shows that all the validation runs except one manage to capture this feature. The run which does not capture it is a single level run at the first level of refinement; it simply does not have the resolution to produce a flat surface at the trailing edge.



(a) MLSO convergence histories.

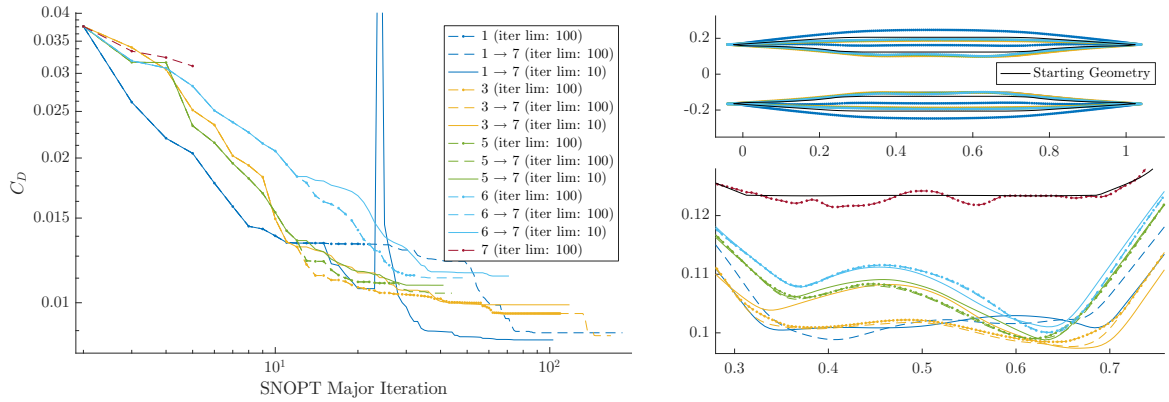
(b) Final profiles (close-up on the trailing edge).

Figure 6.18: MLSO runs for different settings from 1 to 7 for a single body starting geometry.

Figure 6.18a shows the convergence histories of the MLSO validation runs of the single body. These display the stepped convergence that is expected of multi-level parameterisation: each new level of refinement unlocks a portion of the design space, enabling further improvements. The solid lines in this plot were run using only 10 steps at each level, as analysis of the previous section suggested this would be sufficient. This is confirmed, as the final objective was not compromised by this setting: the runs starting at levels 1, 3 and 5 converged in fewer iterations on a similar design by restricting the number of steps at intermediate levels.

While most optimisation runs capture the blunt trailing edge, they do it with varying levels of precision. These discrepancies in geometry are reflected in the final drag values shown in Figure 6.18a. The MLSO appears to struggle to explore this region of the design space, this is likely due to the chord length being under-constrained, grid sensitivity and dependency at the trailing edge, and, inaccurate design sensitivities at that location. The weak chord constraint which allows a 4% error margin, as a result these profiles must be compared to the truncated ogive of area 0.11, which has a drag coefficient of 0.069.

The sensitivities, and therefore the search direction, rely on the adjoint to the flow, however this region displays flow separation despite the use of an inviscid solver. This type of separation is a result of artificial dissipation, and does not accurately model the physical behaviour of the region. As a result, the flow solver does not converge sufficiently reliably in the re-circulatory region to allow a consistent search direction. This is reflected in the residuals of the adjoint: while the overall residuals are converged to 10^{-10} this region of the adjoint exhibits the maximum residual of 10^{-2}. MLSO brings



(a) MLSO convergence histories.

(b) Final profiles (close-up on the inside of the throat).

Figure 6.19: MLSO runs at multi-level subdivision settings from 1 to 7 for a starting geometry composed of 2 bodies.

the flat trailing edge as close to that limit as possible, however due to the inaccurate search direction and the varying level of fidelity, the path followed by the optimisation varies and stalls at different points.

Trailing edges with this blunt geometry also exhibit important sensitivity to mesh density, as shown in Table 6.2 in Section 6.2. While re-meshing is used to avoid badly shaped elements the current MLSO implementation only supports a fixed surface and volume mesh density. Enabling a mechanism for increased resolution in these regions could lead to more consistent optimised profiles.

6.5.1.2 Multi-body profile

The same testing process was used for a starting geometry made of two bodies. This region of the area constraint space is interesting aerodynamically as the Busemann bi-plane is not optimal. If the shock cancellations expected for a Busemann biplane (Figure 6.1a) do not hold, the flow is choked: a large bow-shock forms in front of the geometry causing a step increase in drag for the biplanes at areas above 0.1. To succeed the optimiser needs to balance the choking of the flow with external curved edges which generate shocks that are not cancelled out.

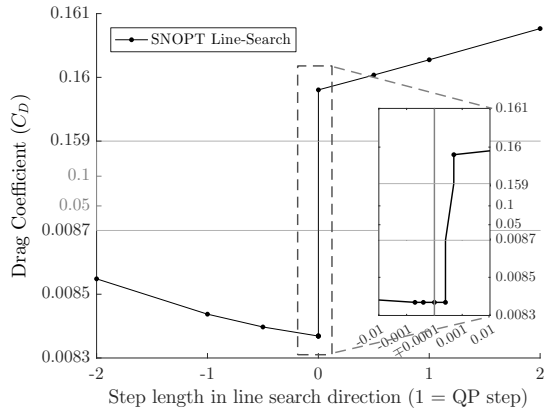
Figure 6.19 shows that all but the seventh single level case successfully shift the external parts of the geometry to the region between the two bodies. This movement in-board reduces the component of wave drag that is not cancelled by interaction with the second body. This behaviour, shown in Figure 6.19a, appears in the initial steep

decrease in drag coefficient down to 0.014.

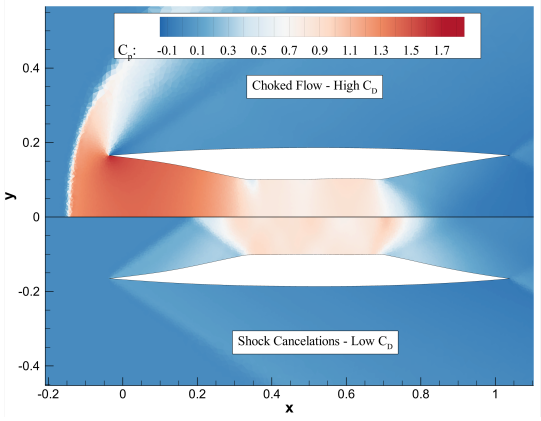
Despite its desirable initial behaviour, the MLSO does not appear to converge reliably on the same drag value. These differences in drag are reflected in the shape of the throat of the profile, shown in Figure 6.19b. The shape of the profile of the throat is responsible for two aspects of the flow: the quality of the shock cancellations and whether the flow chokes. To understand the reason for this discrepancy, the drag values in the line search direction are plotted, in Figure 6.20a, for the best run in Figure 6.19b. This figure shows a very large discontinuity in the drag value along the SQP search direction; C_P plots on both sides of the discontinuity (fig 6.20b) reveal the complete change in flow behaviour responsible for this discontinuity. The reason MLSO stalls when it reaches that point in the design space is that information about this change in shock pattern is not captured by the adjoint and cannot be reflected in the search direction.

This ill-conditioning of the flow solution at the optimum solution is similar to the hysteresis with Mach number observed in the aerodynamics design optimisation discussion group (ADODG) case 1[3] by Destarac et al. [194]. The repeated appearance of this type of issues illustrates a pitfall of single-point, single objective optimisation: the optima generated by these problems tend to be very narrow and unusable in design. This points to this type of optimisation as exploratory methods for designers to use and improve their understanding of what can be done within a set of design constraints.

[3] http://info.aiaa.org/tac/ASG/APATC/AeroDesignOpt-DG

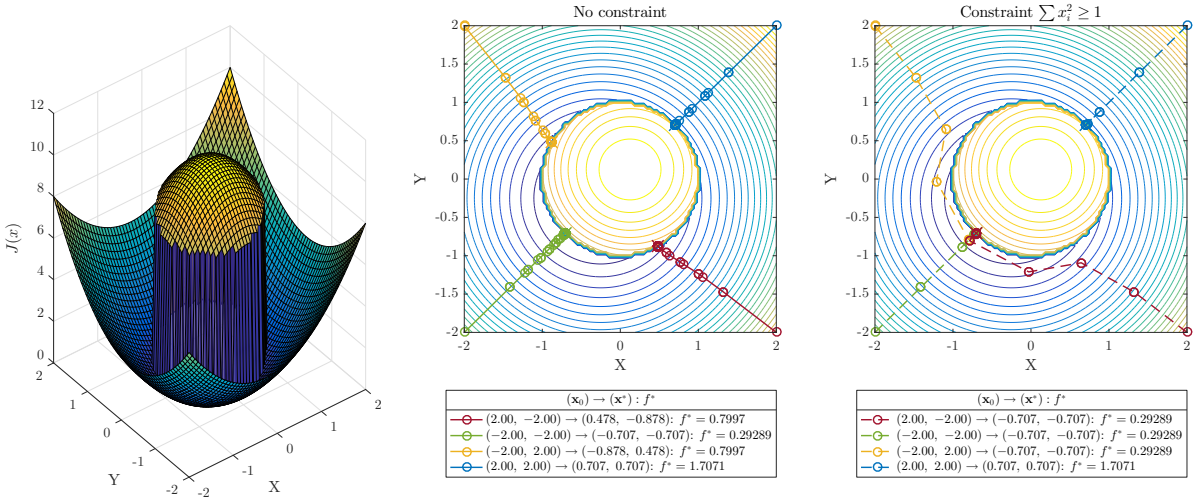


(a) C_D along a line search direction at the end of a MLSO run, a very small step (< 0.001) causes a step increase in drag (×20).



(b) C_P flood plots for the high and low drag flow regimes.

Figure 6.20: Analysis of the aerodynamic design space at the optimum geometry, it lies close to the choke point where the flow changes abruptly.



$$J(\mathbf{x}) = \begin{cases} \frac{1}{2}\sum_i x_i + \sum_i x_i^2 & \text{for}: \quad \sum_i x_i^2 \geq 1 \\ \frac{1}{2}\sum_i x_i - 2\sum_i x_i^2 + 10 & \text{for}: \quad \sum_i x_i^2 < 1 \end{cases} \tag{6.4}$$

Figure 6.21: Impact of objective function discontinuity on optimisation and ability of constraints to circumvent these short comings. The objective function is an analytical replacement for the choking behaviour exhibited by multi-plan profiles.

One unexpected feature in the convergence of the first to seventh level optimisation with 10 iterations is the large step-up in drag at the 20^{th} iteration. This is due to an unexpected interaction between the MLSO and the re-meshing of the geometry; the result of this is a change in the flow regime causing the shape optimisation to continue a previously un-choked solution in the choked regime.

6.5.1.3 Impact of objective discontinuity on optimiser convergence

In its current format the final solution to the multi-body shape optimisation problem is very sensitive to starting geometry and parameterisation settings. This is because the need to reduce the external shocks rapidly drives the geometry to the limit of choking. Because choking is caused by the flow going subsonic it is very sensitive to the shock patterns between the bodies. Small differences in paths through the design space lead to this “choking boundary" to be encountered by different profiles. Once this boundary has been encountered the optimiser cannot progress as the existence of the discontinuity is not reflected in the design sensitivities and therefore the search direction. The current formulation of this case appears to be multi-modal when considering shape

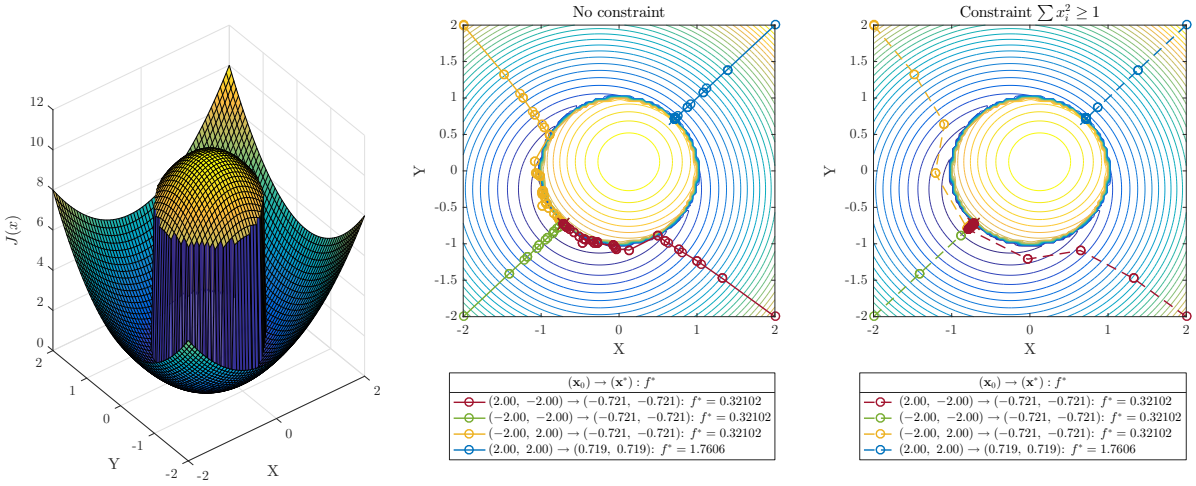
only; additional information would be needed to allow the optimiser to progress beyond the choking boundary.

The behaviour engendered by a choking boundary can be simulated using the analytical function presented in Equation 6.4, in Figure 6.21. The underlying parabolic function is discontinuous in a circle of radius 1 around the origin and exhibits the same line-search properties exhibited by the choking boundary in Figure 6.20. This analytical function has a single (global) optima at $(-1/\sqrt{2},\ -1/\sqrt{2})$. For the unconstrained discontinuous case, the optimiser is unable to circumvent the discontinuity to find an optima despite the existence of a continuous monotonic descent path. Introducing a non-linear constraint along the discontinuity allows the optimiser to move around, finding the optima.

A constraint for choking discontinuity could manifest itself in two ways: a physical constraint or an algorithmic one. A physical constraint may be possible as choking is the result of the flow reaching Mach 1 at the throat of the profile, the challenge with this approach is that it requires an additional flow sensitivity which must be implemented into the flow solver to maintain the benefit of using adjoints. An algorithmic constraint approach would entail the dynamic addition of a constraint to SNOPT which matches the perceived boundary of the discontinuity. Upon failure of the optimiser and detection of a discontinuity, an additional constraint be introduced indicating to the optimiser that a region of the design space must be avoided. This constraint can be a simple quadratic constraint on the design variables ($\mathbf{x}$), forming a (hyper-)spherical constraint lying where the discontinuity is detected (Equation 6.6). In Equation 6.6, $\mathbf{x}_d^*$ is the 'optimum' found by the optimiser due to a discontinuity, $\widehat{\Delta\mathbf{x}}_d$ is the last line search direction on which the discontinuity was encountered, and σ is a tuning parameter (superior to 0) controlling the size of the constrained region of the design space.

$$\text{Detected discontinuity constraint}: \quad \sigma - \sum_i \left[\mathbf{x} - \left(\mathbf{x}_d^* + \sigma\widehat{\Delta\mathbf{x}}_d\right)\right]^2 \leq 0 \tag{6.6}$$

Another potential method to circumvent the discontinuity would be to use a local continuous surrogate model of the objective on which the gradient based optimisation is performed. In Figure 6.22 the discontinuity is replaced with a very steep continuous hyperbolic tan barrier. Using this objective instead of the discontinuous one, the unconstrained minimisations find the global minimum at the same rate as the constrained cases. This phenomenon indicates that a method performing the optimisation on a continuous local surrogate of the objective would likely be successful. While beyond the scope of the current work an existing or a new surrogate based optimisation method could successfully explore the discontinuous region of the design space.



$$J(\mathbf{x}) = \sum_i x_i + \sum_i x_i^2 + \left(10 - 3\sum_i x_i^2\right)\left(\tanh\left(100\left[1 - \sum_i x_i^2\right]\right) + 1\right) \tag{6.5}$$

Figure 6.22: Impact of using a continuous switch to mimic the discontinuity, the unconstrained optimisation is now well behaved, suggesting surrogate based optimisation could avoid the stalling behaviour caused by discontinuities.

These findings show that the failure of SNOPT to find minima are not necessarily the result of multi-modality. Three approaches have been outlined as potential solutions to optimise around the discontinuity caused by the change of flow topology. Although they have not been implemented, evidence has been presented which suggest they could be effective solutions. Based on current understanding of the author the most logical approach to follow is that of an algorithmic constraint, removing regions of the design space once discontinuities are detected. It is relatively simple to implement with few foreseeable side effects and is the most general: it can be applied as a black box solution to any objective function.

6.5.2 Comparison to Previous Aerodynamic Topology Results

The data used to validate the convergence behaviour of the combined shape and topology optimisation framework have been presented, also highlighting some potential pitfalls in the optimisation cases. In this section, the performance of the combined MLSO-RSVS framework is compared to earlier results using only the RSVS on the same test cases. The combined framework is compared to the analytical results discussed in Section 6.3 and the optimisation results generated by each of the MLSO and RSVS methods in

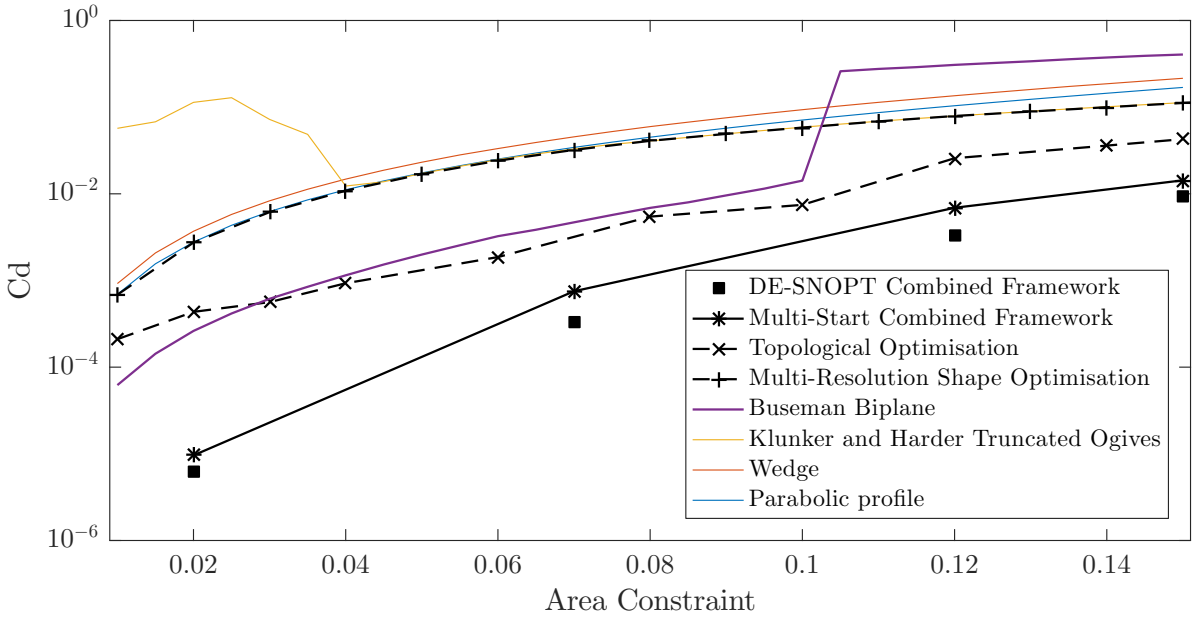


Figure 6.23: Comparison of the combined shape and topology optimisation framework to the linear theory optima and the results of normal ASO and ATO frameworks.

Figure 6.23.

Figure 6.23 shows the performance improvement enabled by the combined framework. The combination of the RSVS geometry representation with the multi-level subdivision shape optimisation framework allows consistent improvements in optimum drag through the range of volume constraints. The performance of the combined framework is further improved by starting the shape optimisation from a population which has been explored using DE and the RSVS. This approach of sequential optimisation lets DE generate high performing aerodynamic topologies while replacing the extremely long and slow convergence of the global optimiser with efficient MLSO runs on each member of the population. These improvements show the additional capability of the framework compared to the RSVS and MLSO; it opens up the use of the combined method to tackle more complex aerodynamic topology optimisation problems.

The full hybrid optimisation method (Figure 6.14c) was not competitive in its current form: it performed no better than the multi-start algorithm and was more expensive. A detailed inspection of intermediate results suggests that many geometries generated by the RSVS do not capture meaningful aerodynamic features and are poor candidates for MLSO. While generation of such profiles is the natural behaviour of the DE combination process; it does not allow objective function improvements compared to the multi-start process, in particular when the DE population size is reduced such that both algorithms

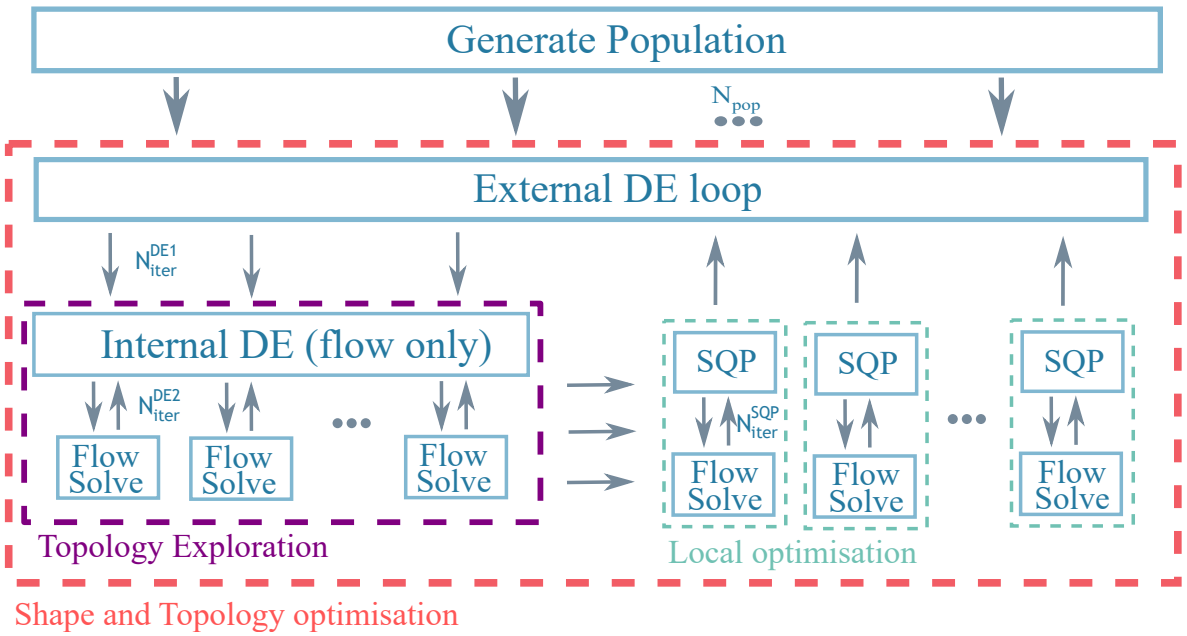


Figure 6.24: Proposed hybrid optimiser with flow-only DE steps followed by shape optimisation using the MLSO framework.

have similar wall-times.

Figure 6.24 proposes an alternate implementation of a hybrid DE-MLSO method which would be expected to improve performance. The method of Figure 6.24 was not tested and is suggested as a means of tackling problems with more complex constraints. It was felt that the multi-start and sequential approaches were sufficiently thorough in their exploration of the design and constraint space for the problems tackled in this work.

6.5.3 Study of Optimisation Cases with Additional Topological Flexibility

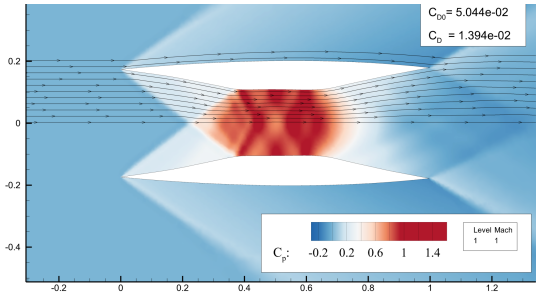
Having shown the capability of the hybrid framework on single and multi-body profiles, additional topological flexibility was added to the design space. This is achieved by increasing the resolution of the RSVS in the vertical direction. For previous cases the RSVS design variables were in a 6 by 10 layout, while these new cases are generated using a 2 by 20. The expectation was that this additional flexibility will allow the multi-start and the sequential optimisation processes to compare the performance of different types of multi-plane profiles more thoroughly. The area constrained drag minimisation case presented in Section 6.5.1 is tackled for a constraint value of 0.12.

The first challenge introduced by the additional topological flexibility is the generation of a good starting population. The quality of this population impacts both the multi-start gradient based algorithm and DE. For effective global optimisation a starting population will need sufficient diversity of meaningful aerodynamic profiles. Either of these properties on its own is not sufficient: to adequately explore a fully random population a prohibitive number of agents would be required; and without diversity global optimisation will not be able to generate sufficiently varied designs.

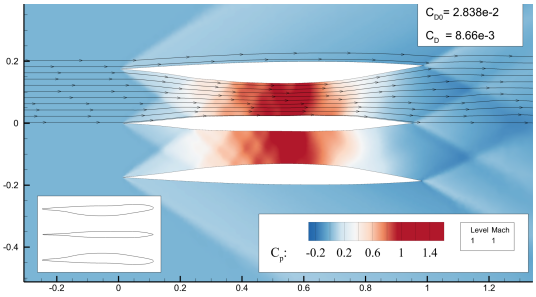
The refinement of the design space in the normal direction of the flow led to poor performance in the starting population: many more of the generated profiles are in the choked flow regime. Choking of the flow by far dominates the drag performance of a geometry. In addition, this disproportionately affects profiles made of more bodies. This limitation of the starting population causes the failure of the differential evolution as it rapidly converges on the non-choked profiles that appear most commonly despite their poor drag performance compared to results obtained in previous sections. Significantly increasing the size of the population would allow the starting population generation to generate better profiles and increase the chance of good multi-planes to appear in the population before premature convergence can set in.

Despite the poor aerodynamic performance of the starting population, the multi-start algorithm highlights some interesting behaviours. The severe discontinuity in the aerodynamic design space caused by the change in flow topology poses a challenge: there is no guarantee that drag reduction on the 'high drag' side of the discontinuity will reliably guide the optimiser towards the desired change in flow topology. This is especially true for profiles with a more complex topology because of the increased geometric complexity of the local design space at the lowest subdivision level. This can cause the optimiser to follow design directions away from the low drag behaviour allowed by shock cancellations. These directions manifest themselves in the very different final geometries generated by the local optimiser from similar starting profiles.

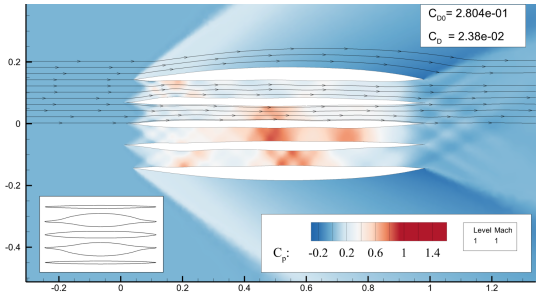
Cases are presented in Figure 6.25 for 5 and 6 plane profiles. The best and worst cases presented (figs 6.25c and 6.25g respectively) show extremely similar starting profiles with drastically different optimised results. Comparing Figures 6.25c and 6.25d two profiles with the same starting topology but different geometries converge on profiles relying on very different shock interactions to achieve a low drag value. This suggests multi-modality of the optimisation problem even for a given topology. Interestingly, Figure 6.25h shows a profile with a different topology (6 bodies) but tending to a flow pattern similar to Figure 6.25d. While mostly of academic interest, these cases resemble



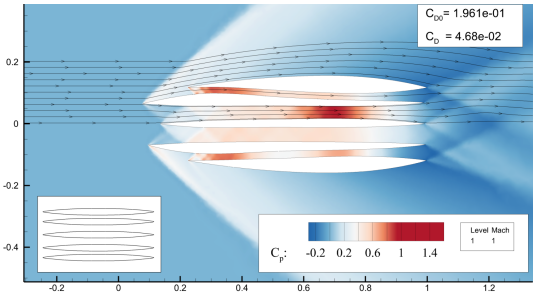
(a) 2 body profile



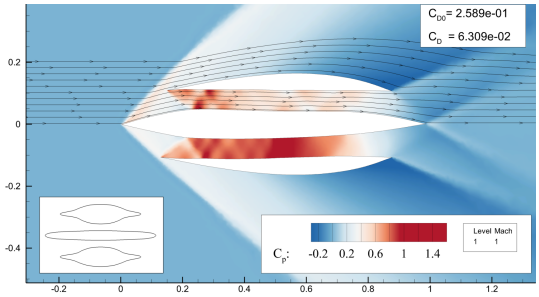
(b) Best 3 body profile



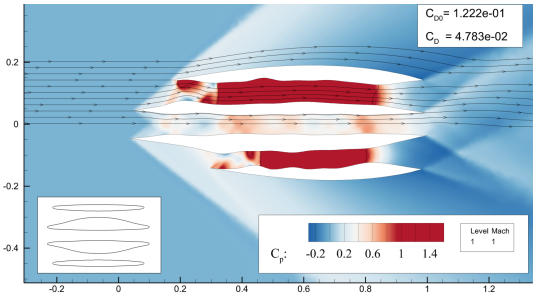
(c) Best penta-plane



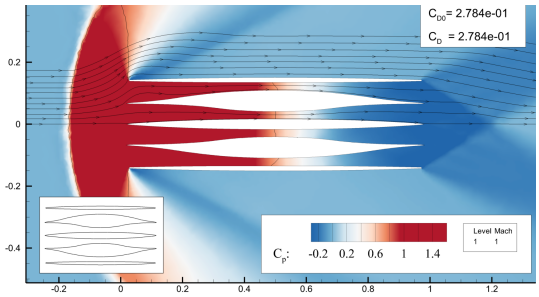
(d) Successfully optimised penta-plane with a different flow topology



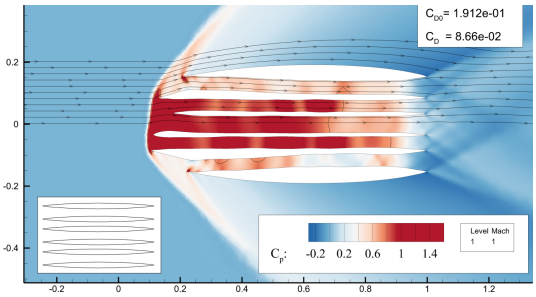
(e) Locally optimal tri-plane with the second flow topology



(f) Quad-plane with the second flow topology



(g) Failed optimisation.



(h) Partially Optimised profile

Figure 6.25: Pressure coefficient flood plots for profiles optimised by the MS-MLSO framework on a population with increased topological flexibility for a $c_A = 0.12$. In each subfigure the initial drag (C_{D0}) and the final drag (C_D) is stated. The starting profile is inset at the bottom left of the images. The profiles selected highlight the geometric and aerodynamic topology properties which exist for local minima of the optimisation case.

the shapes of supersonic engine inlets; where shocks are used to perform parts of the compression.

These observations suggest two possible approaches for subsequent studies of this class of optimisation cases: continued search for lower drag; or an attempt to find and classify the local optima that are discovered. A fully hybrid framework, where DE takes into account gradient information to rank its agents, would make up for the shortcoming of the individual optimisers and would lead to better solutions than those found so far for the cases involving more topology. Indeed, by including a measure of the 'optimisation potential' of the geometry, the global optimiser, instead of optimising for drag, will optimise the population for its suitability for the MLSO process.

Observations of the solutions reveal that both flow topology and geometric topology affect the modality of the optimisation independently. The multiple minima engendered by changes of geometric topology were expected; differences due to flow patterns are not surprising either. What is surprising is the similarities in flow patterns for optima which have different geometric topologies; these suggest that regardless of topology the optimiser taps into the same underlying aerodynamic processes to achieve a low value of drag. The flow similarity is evident in the shock cancellations appearing in Figures 6.25a, 6.25b and 6.25c, despite having two, three and five bodies respectively. The second flow topology which was identified relies on shorter external bodies which do not cancel out the shocks generated by the inner profiles but rather match the angle of the flow behind the shock. This behaviour is seen in Figures 6.25e, 6.25f and 6.25d; which have, respectively, 3, 4 and 5 bodies.

The differences between these two groups of optima shows how the integrated RSVS-MLSO approach allows the exploration of geometric and aerodynamic design space. While in traditional 'one-shot' optimisation, the chaotic response of the MLSO to small changes of initial geometry (as seen in Figure 6.25) would be consider a failure; in this context it is revealing of the underlying properties of the design space. By providing evidence for the discontinuity in the aerodynamic design space, it builds the understanding of the designer of what is possible within the constraints of a specific problem. While the current framework is effective at revealing the properties of design space, the algorithms used are tuned for minimisation: significant efficiency gains could be made by using algorithms developed specifically for exploration tasks.

The high level of multi-modality both in terms of flow behaviour and geometric topology makes this case ideal for niching and quality diversity [259] approaches. A recent study using a niching variant of differential evolution has successfully been used to

identify multiple minima during the optimisation of a wing [82]. These methods instead of looking at convergence on a single local optima aim to return a number of locally optimal solutions. These allow the designer to gain a better understanding of the design space available and can be used at earlier stages of design to suggest design directions. These could be particularly interesting when looking at complex sets of constraints interacting with the topological freedom afforded by the RSVS. Section 6.5.1 discussed a severe discontinuity of the aerodynamic design space due to "choking" of multi-plane profiles(first published by the authors in Kedward et al. [248]). Niching could provide an effective approach in the exploration of multi-modality around this discontinuity and diversity due to changes of the constraints.

6.6 Summary

The set of cases explored in this chapter compared the effectiveness of shape optimisation to the potential of topology optimisation. The optimisation cases were tackled parametrically: with a single framework configuration, a suite of case with varying geometric and flow properties were tackled. The use of the MLSO in combination with the RSVS enabled faster and more thorough exploration of the design space than previously possible: navigation of both geometric and flow topology was achieved.

This combination of tools enabled the qualification of aerodynamic and geometric features of the design space; highlighting in particular cyclical multi-modality with regard to topology, and a possible degeneracy close to optima in inviscid supersonic flows. Like the optimised NACA0012, the flow is extremely sensitive to flow conditions and geometric changes. These observations confirm the notion that single-point inviscid drag minimisations tend to degenerate designs with very narrow windows of performance. While Reynolds-Averaged Navier-Stokes (RANS) based optimisation has been found to alleviate some of these behaviours, the need for robust unsupervised meshing could not be satisfied with the tools available to us. The very high sensitivity of drag optimised aerofoils, means that the result of single-point, Euler, drag minimisation cannot be used on its own in the design process and must be enhanced with other sources of information.

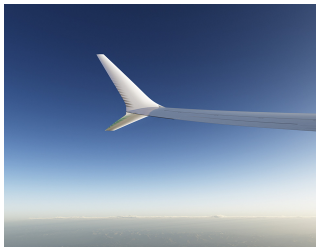
CHAPTER 7

EXTENSION OF THE RSVS TO THREE DIMENSIONS

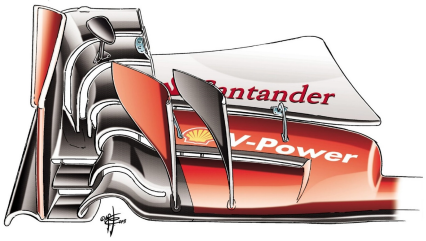
While there exists a wide range of parameterisation methods for aerodynamics in two dimensions, in three dimensions the parameterisation offering is dominated by free-form deformation (FFD) methods. Thanks to their deformative formulation, an existing discrete geometry can be used as the starting point, maintaining its properties. By preserving surface characteristics, deformative methods allow a high level of accuracy provided the initial geometry is of high quality; however this precludes topological design of aerodynamic bodies. Generating the initial geometries and discretisations is a significant challenge in its own right, with the water-tightness of surfaces not generally guaranteed by the CAD tools used to define them.

These challenges mean that aerodynamic topological optimisation of an entire aircraft or wing is unlikely to be a reality in the near or medium term. However, there is scope for the aerodynamic topological design of local features; topological optimisation of wing tips would allow feathered or split winglets more complex than that on the Boeing 737-MAX to be explored (Figure 7.1a and Figure 7.1c). An effective topological aerodynamic optimisation framework also offers the possibility of radically new designs in applications to: Formula 1 (Figure 7.1b), unmanned aerial vehicles, commercial strut-braced wing design, and, internal engine design. No current optimisation framework for external aerodynamics supports the exploration of 3-dimensional topological changes, in large part because of the dominance of deformative parameterisation methods.

The area constrained length minimisation formulation of the restricted snakes volume of solid (RSVS) offers a natural extension to 3D as the minimisation of the surface area of a geometry under volume constraints. During design and testing of the two dimensional



(a) Winglet of the Boeing 737-MAX, from Boeing [17].



(b) Front wing of the SF-15 Ferrari F1 car, from Giorgio Piola [18].

(c) Eagle in flight.

Figure 7.1: Examples of complex topology in aerodynamic applications.

RSVS; features which would extend to 3D were prioritised. The flexibility and generality of the RSVS formulation, useful features in two dimension, becomes necessary for efficient exploration of 3 dimensional design spaces where isotropic tessellation of the geometric space would be prohibitively expensive.

A new ‘restricted surface (r-surface)’ tool is developed, generalising the restricted snake (r-snake) to higher dimensions, this new method allows the robust evolution and containment of objects of arbitrary topology. This r-surface is then integrated with a marching procedure minimising the area of the surface under localised volume fraction constraints to form the restricted surface volume of solid (3D-RSVS) parameterisation. Properties of this new parameterisation are discussed, highlighting similarities with the generation of constant mean curvature (CMC) surfaces and minimal surfaces. Design of volume of solid (VOS) layouts is treated in Section 7.4.2, and finally the 3D-RSVS is integrated into an optimisation framework.

7.1 Restricted Surfaces Volume of Solid for 3-Dimensional Aerodynamic Parameterisation

This section presents how the 3-dimensional restricted surface volume of solid (3D-RSVS) parameterisation translates sets of volume fraction design variables specified on a fixed grid into closed surfaces of varying topology. For optimisation frameworks to exploit the 3D-RSVS efficiently, this process must reliably produce smooth features at a resolution below the grid on which VOS values are defined. To achieve this level of smooth control, the 3D-RSVS profile is defined as: the closed surface of minimum area that will match the volumes of the design variables. It is built using a restricted surface (r-surface). The r-surface is a method developed in this thesis for “vertex marching" which allows efficient

topology handling and is tolerant of any layout of VOS design variables. The 3D-RSVS is implemented in C++, the code is made available by the author on GitHub[1] under the LGPL-3.0 license[2].

7.1.1 Governing Equations of the 3D-RSVS

By using a formulation which is very closely related to the two dimensional parameterisation, the 3D-RSVS maintains many of the benefits of that parameterisation, notably: intuitiveness, homogeneity, smoothness, compactness and flexibility. The parameterisation process is shown in Figure 7.2 with slices through the domain indicating the volume fractions.

The 3-dimensional RSVS geometries are defined as the surface with the smallest area matching the VOS in every cell. The mathematical formulation of this problem is given in Equation 7.1. This system is analogous to the effect of a tensile force “shrink-wrapping" the required VOS in each cell. The general form of the 3-dimensional RSVS problem is developed for a closed surface S which is constrained in m design cells (C_j) to have a specified volume fraction V_j. These variables are represented graphically in Figure 7.2b for a 2 dimensional grid.

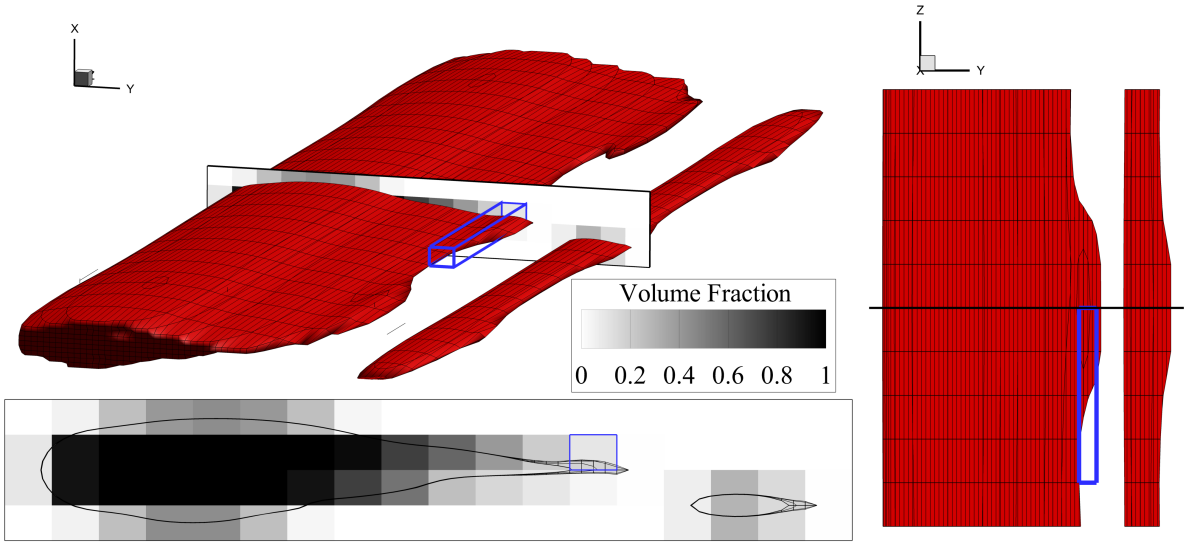
$$\begin{aligned} \min \quad & \iint_{S \to \mathbf{x}(t,u)} \left\| \frac{\partial \mathbf{x}}{\partial t} \times \frac{\partial \mathbf{x}}{\partial u} \right\| dt\, du \\ \text{s.t.} \quad & \iiint_{(S \cap C_j)} dx\, dy\, dz = V_j \qquad \forall j \in \{0, \cdots, m\} \end{aligned} \tag{7.1}$$

The rules above are the natural extension to 3-dimensions of the 2D-RSVS: the length minimisation has become a surface minimisation and the area constraints become volume constraints. The design variables that control the surface are volume fractions specified in each cell of a *design grid* which remains unchanged during an aerodynamic topology optimisation procedure. The next sections detail how the mathematical program is solved using restricted surfaces to produce an effective shape and topology parameterisation method.

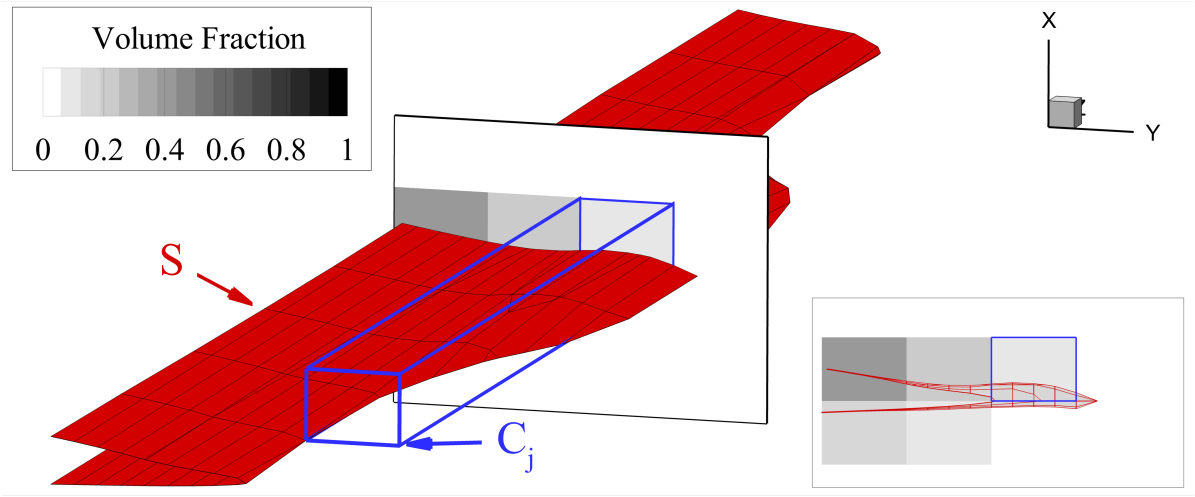
Unlike the two dimensional case; the 3D-RSVS does not support exact analytical solutions. The 3D-RSVS surfaces are part of a class of problems known as constant mean curvature (CMC), a super-class of “minimal surfaces”, for which explicit analytical

[1] `https://github.com/payoto/rsvs3d` and `https://github.com/farg-bristol/rsvs3d`

[2] Available at: `https://opensource.org/licenses/lgpl-3.0.html` accessed on 05/06/2019.



(a) VOS design variables as grey-scale and 3D-RSVS surface; 1 corresponds to a completely full cell and 0 an empty cell.



(b) VOS definitions for Equation 7.1

Figure 7.2: Example RSVS profile and design grid with label definitions for the governing equation (Equation 7.1).

solutions are available for specific boundary conditions [260]. In the RSVS process the boundary conditions, as understood in the study of minimal surfaces, are not known explicitly as they are the result of the marching of the snake or surface. This additional level of complexity makes general analytical solutions to the 3D-RSVS unlikely; however comparison of specific parameterised layouts to existing analytical solutions is performed in Section 7.2. A more substantial discussion of minimal surfaces and CMC surfaces is done in Section 7.4.1.

7.1.2 The Restricted Surface Method for 3D Topology Evolution

Development of a 3-dimensional, volume of solid based, topologically flexible parameterisation requires an efficient method for evolving topologically complex geometries. The 2D-RSVS used restricted snakes, a type of parametric active contour developed by Bischoff et al. [217, 231]. Previous work in the extension of parametric active contour methods to 3 dimensions have been successful, notably the development of topologically flexible T-surfaces by McInerney et al. [215] for medical image segmentation. This subsection outlines the extension of the r-snake to evolve as a surface on 3-dimensional grids, and how this may be used to solve the 3D-RSVS governing equation. A complete description of the development of these r-surfaces is provided is Section 7.3.

7.1.2.1 Topology evolution of polyhedra

To build the 3D-RSVS parameterisation method the restricted surface must be evolved until it solves the governing equation. The restricted surface is a vertex marching procedure where the control points (called snaxels) are constrained to move on a predefined grid, as a consequence properties of the snaking grid controls the number of snaxels and the resolution of the geometry. By marching the snake on a grid finer than the VOS grid, smooth features below the resolution of the volume design variables can be recovered. This allows a high degree of geometric flexibility with few design variables.

In order to maintain the water-tightness of the surface the connectivity elements between snaxels, are restricted. The original rules developed by Bischoff and Kobbelt [217] for contours have been generalised to surfaces in 3D space into the following:

- No 2 snaxels connected by a r-surface edge are on the same *snaking grid* edge;
- No 2 r-surface edges connected by a snaxel are in the same *snaking grid* face;
- No 2 r-surface faces connected by r-surface edge are in the same *snaking grid* cell.

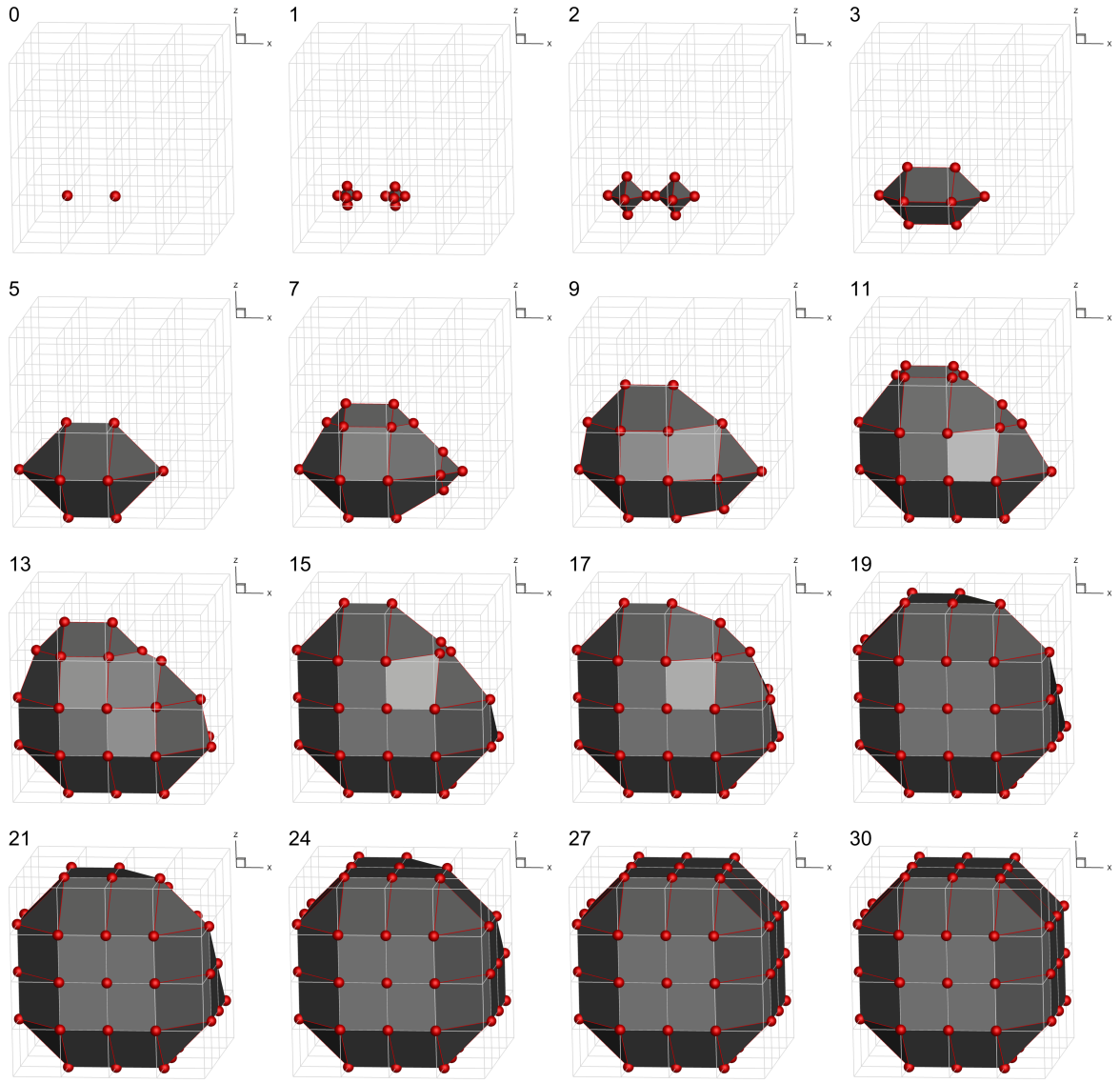


Figure 7.3: Evolution of a restricted surface in a 4^3 snaking grid spawned in two locations under a unit velocity field.

These connectivity rules, can be used to maintain a meaningful surface when two surfaces collide or when a surface crosses through a vertex of the snaking grid. Figure 7.3 shows the evolution of restricted surfaces initialised from two vertices in a 4^3 snaking grid. The two surfaces collide on the third step, and the connectivity is adjusted using the rules specified above.

While the rules for building the r-surface guarantee the formation of water-tight surfaces, it does not guarantee that faces will be flat. This is because the r-surface is controlled by the positioning of its vertices with the rest of the geometry derived from

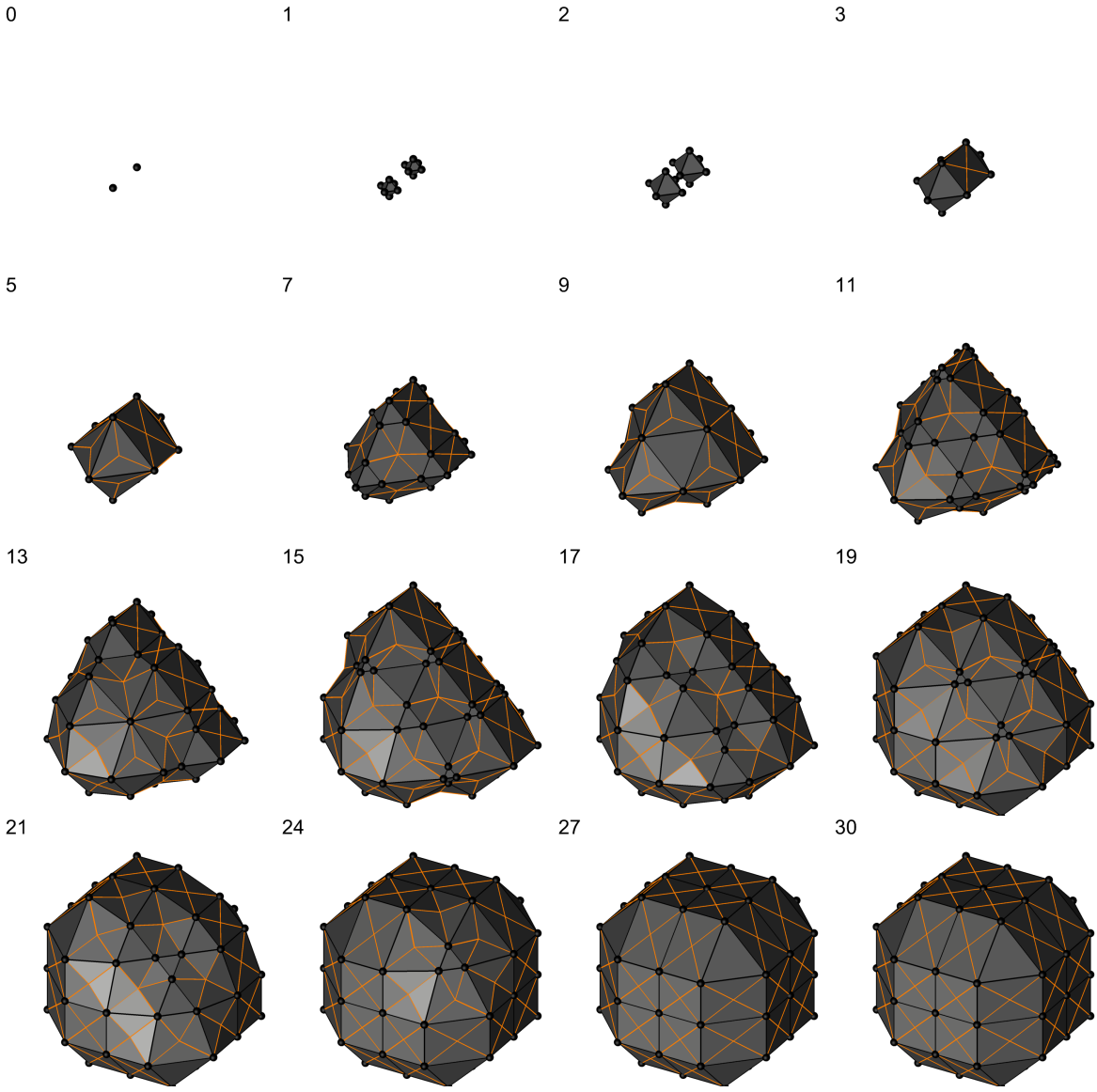
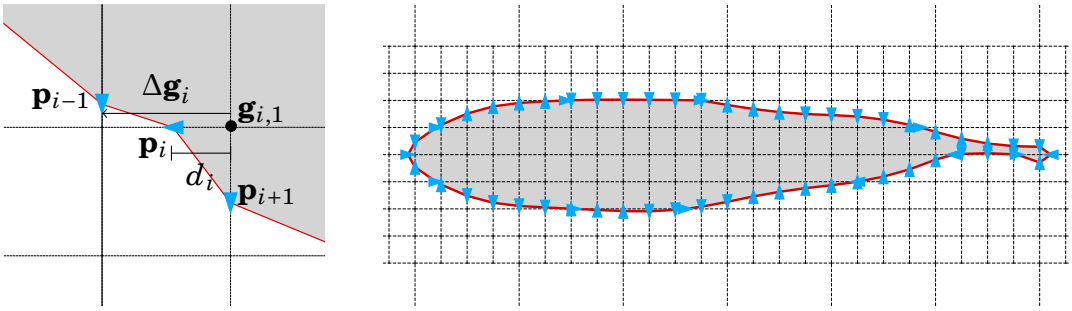


Figure 7.4: Evolution of the triangulation (edges in orange) of a restricted surface (in black) using the centroid defined in Equation 7.3. Step numbers are in the top left corner.

the connectivity information forced by the snaking grid. Flat surfaces are required for the reliable calculation of volume and area of the polyhedron, faces with more than 3 edges need to be triangulated. Consistency and smoothness of the triangulation through changes of connectivity is achieved by triangulating faces through point $\bar{\mathbf{c}}$; this point is the mean position of face vertices normalised by edge length. Figure 7.4 shows the evolution of the chosen triangulation process through the evolution of the restricted surface.



(a) Close-up view of snake and snaxel

(b) Full view of a closed r-snake

Figure 7.5: R-snake contour (in red) with snaxels (in blue) evolving on the snaking grid (dashed line) [Repetition of Figure 3.9].

7.1.2.2 Integration of the r-surface with the area minimisation problem

To drive the position of the restricted surface the original continuous area minimisation problem (Equation 7.1) is discretised in terms of the r-surface and snaxel variables, becoming the mathematical program of Equation 7.2. The discretisation process expresses the integrals in terms of the triangulated r-surface using six properties of the r-surface geometry and the snaxel positions. The first three of these properties are part of the movement algorithm; the last three properties of the snaxels are derived from connectivity and grid information. These properties are: the snaxel index (i), used to reference it in all operations; the normalised position along an edge ($d_i \in [0,1]$); the scalar velocity along that edge ($v_i \in \mathbb{R}$); the snaxel position in Cartesian coordinates ($\mathbf{p}_i$); and; the direction of travel of the snaxel ($\Delta\mathbf{g}_i$) and the vertex of origin ($\mathbf{g_{i,1}}$). These properties are represented graphically for a restricted snake in Figure 7.5a. They are used to calculate and differentiate $A_{S,k}$ and $V_{S,k}$, respectively the area and volume contributed by each triangle forming the polyhedron. These definitions are integrated into Equation 7.1 to form the discrete mathematical program used to drive the r-surface in Equation 7.2.

$$\begin{aligned} &\min_{\mathbf{d}} \sum_{k=1}^{q} A_{S,k}\left(\mathbf{p}_{0,k},\mathbf{p}_{1,k},\mathbf{p}_{2,k}\right) \quad \text{with } \mathbf{p}_{i,k}(\mathbf{d}) \\ &\text{s.t.} \sum_{k=j_S(1)}^{j_S(q_{j_S})} V_{S,k}\left(\mathbf{p}_{0,k},\mathbf{p}_{1,k},\mathbf{p}_{2,k}\right) + \sum_{k=j_C(1)}^{j_C(q_{j_C})} V_{C,k}\left(\mathbf{g}_{0,k},\mathbf{g}_{1,k},\mathbf{g}_{2,k}\right) = V_j \end{aligned} \tag{7.2}$$

Building an RSVS surface requires the positions $\mathbf{d}$ of the r-surface snaxels satisfying Equation 7.2 to be found. As is the case in 2-dimensions the objective function and the

constraint are readily differentiable. This is critical to solving the area minimisation governing equation as it allows the use of efficient gradient based optimisation method. While the area and volume could be differentiated by hand, the task would be tedious and error prone. The differentiation of $A_{S,k}$, and $V_{S,k}$ with regard to $\mathbf{p}_{i,k}$ was carried out for triangles using the MATLAB symbolic toolbox. This allows C code to be directly generated for the mathematical functions, ensuring that no mistake is made when calculating Jacobian and Hessian. The same process is followed for the derivatives of $\mathbf{p}_{i,k}$ and $\bar{\mathbf{c}}$ with regard to $\mathbf{d}$.

This formulation has the benefit of being very general, it can be tackled on an arbitrary volume grid with any underlying snaking grid with any optimisation method. This generality guarantees a high degree of flexibility in the range of shapes that can be represented. Later sections will show how the r-surface is implemented and how the 3D-RSVS parameterisation can be constructed using a Newton step sequential quadratic programming (SQP) procedure. The next section shows parameterisation results using the 3D-RSVS on Cartesian grids and an empirical study of the behaviour of the parameterisation.

7.2 3 Dimensional Parameterisation Results

This section presents profiles generated using the Restricted Surface method driven by SQPsolving the area minimisation, volume constrained, governing equation. The VOS values are manually specified; these results serve to validate the implementation of the 3D-RSVS, and to highlight the topological flexibility of 3D-RSVS on small layouts of Volume of Solid cells.

7.2.1 Practical Surface Generation

The RSVS rules only specify how to evolve a surface but provide no guidance regarding the initialisation. For aerodynamic applications and more generally the design of external boundaries it is effective to start at the faces which touch a void and a non-empty volume cell. This provides fast convergence and intuitive behaviour to a designer. Internal voids can then be created if the restricted surface has failed to explore non-full volume cells.

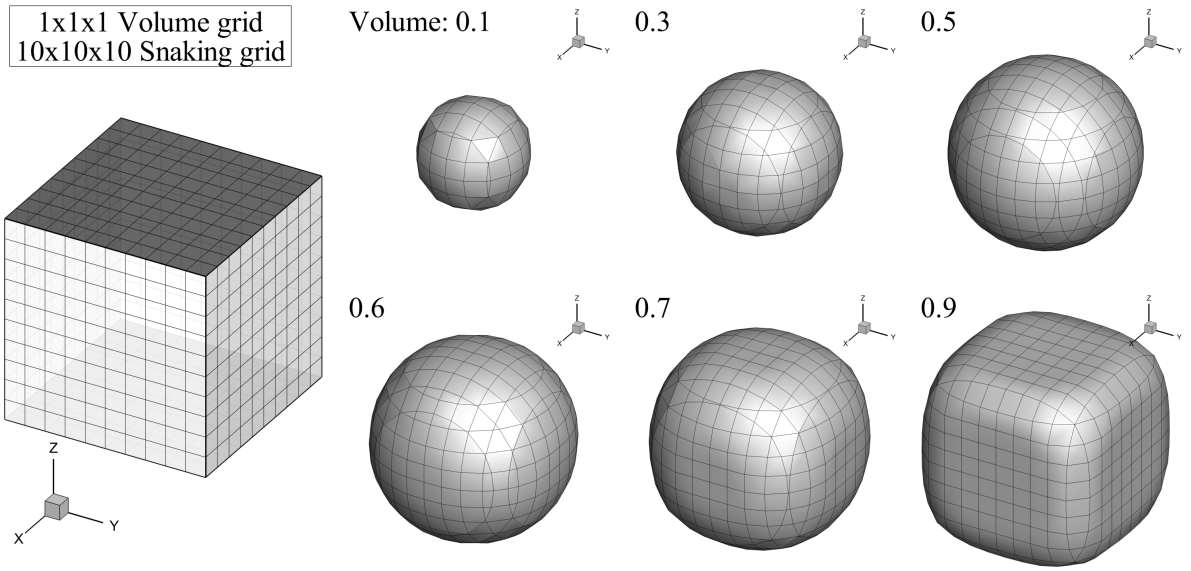


Figure 7.6: Geometries defined by a single volume cell with VOS values from 0.1 to 0.9.

7.2.2 Validation of the RSVS Parameterisation

While an analytical study of the 3D-RSVS problem has not yet been performed it is natural to expect its behaviour to be similar to the two dimensional parameterisation: for shapes defined by few design variables the geometry is hypothesised to be spherical patches. To validate the implementation of the RSVS shapes designed with a single VOS cell were generated and are shown in Figure 7.6. This single volume cell is refined into a 10^3 snaking grid. These shapes clearly show that the RSVS converges to spherical profiles up to volume fractions of 0.5. Beyond that volume fraction, the surface comes in contact with the VOS cell boundaries and starts to form a cuboid with round edges.

To supplement the qualitative observations from Figure 7.6, the volume and area of 3D-RSVS bodies is compared to spheres of equivalent volumes in Table 7.1. This table shows that for low values of requested volume fraction (up to 0.5) volume convergence is good ($\lessapprox 10^{-5}$). As the required VOS approaches 0.5 the area approaches that of a sphere, the area error dropping as low as 0.34% for a sphere of volume 0.5. This is expected: the discretisation of the sphere produced by the 3D-RSVS depends on the number of intersections the geometry has with the background snaking mesh. As the object gets smaller, the number of intersections reduces and the discretisation becomes worse. This observation is confirmed by generating a sphere of volume 0.5 on a finer snaking grid with 24^3 cells; on this snaking grid the area match was even closer at 0.09% (Table 7.1).

For objects coming in contact with the edges of the design space volume convergence

Table 7.1: Numerical comparison of the areas and volumes of 3D-RSVS geometries and spheres of the same target volume.

Fig.	Expected Sphere Properties			RSVS geometry		Error		Observation
	V	Diameter	Area	Volume	Area	Volume	Area	
7.6	0.1	0.576	1.042	0.100	1.054	3.46E-13	-1.16%	
	0.2	0.726	1.654	0.200	1.672	3.23E-06	-1.08%	
7.6	0.3	0.831	2.167	0.300	2.176	-2.03E-07	-0.40%	
	0.4	0.914	2.625	0.400	2.638	2.62E-05	-0.48%	
7.6	0.5	0.985	3.046	0.500	3.057	-2.45E-07	-0.34%	
7.6	0.6	1.046	3.440	0.599	3.456	1.02E-03	-0.44%	at border
7.6	0.7	1.102	3.813	0.700	3.871	3.52E-04	-1.54%	at border
	0.8	1.152	4.168	0.798	4.362	2.03E-03	-4.66%	at border
7.6	0.9	1.198	4.508	0.893	4.840	8.28E-03	-7.36%	at border
7.10	0.75	0.895	5.030	0.750	5.052	2.70E-04	-0.45%	3DVs 2 spheres
	0.5	0.985	3.046	0.500	3.049	-1.04E-06	-0.09%	24^3 snaking grid

is due to the different treatment of snaxels at the edge of the design space. Indeed these cannot be treated as normal design variables for the area minimisation process as they cannot move further outwards but still must be free to move back inwards. A change to the solver of the quadratic program might be needed to support inequality constraints for those snaxels which can only move in one direction. Approaches similar to QPOPT (the internal quadratic solver of SNOPT) [50] are being investigated to improve convergence speed.

7.2.3 Generation of Shapes of Aerodynamic Interest

With the implementation of the 3D-RSVS equations validated, manually specified aerodynamic surfaces were generated as a test for the smoothness of the geometries produced by multiple design variables. The surfaces chosen were the Sears-Haack body, the truncated Sears-Haack body and a wing with aerofoil cross-sections.

Figure 7.7 shows a Sears-Haack body, and the truncated Sears-Haack body is presented in Figure 7.8. These surfaces use a [10, 2, 2] layout of VOS cells and 4^3 snaking refinement. As was the case in 2 dimensions the RSVS produces mostly smooth profiles but can be forced to produce a sharp corner or a sharp edge by using small volume fractions, providing accurate positioning of the leading and trailing edges. In Figures 7.7 and 7.8, the final volume error is displayed on the background volume mesh; showing that the 3D-RSVS process can very precisely match the volume fractions ($e_v \in [10^{-13},\ 10^{-6}]$).

Drag minimisations of wings are common cases within the aerodynamic shape optimi-

Table 7.2: Numerical comparison of the areas and volumes of 3D-RSVS geometries to spheres of the same target volume.

Figure	Design space size	Design variables	VOS layout	Volume fractions	Expected Sphere Properties			RSVS geometry		Error	
					V	Diameter	Area	Volume	Area	Volume	Area
fig. 7.6 up-le	[1, 1, 1]	1	[1, 1, 1]	0.1	0.1	0.576	1.042	0.100	1.054	3.46E-13	-1.16%
	[1, 1, 1]	1	[1, 1, 1]	0.2	0.2	0.726	1.654	0.200	1.672	3.23E-06	-1.08%
fig. 7.6 up-mid	[1, 1, 1]	1	[1, 1, 1]	0.3	0.3	0.831	2.167	0.300	2.176	-2.03E-07	-0.40%
	[1, 1, 1]	1	[1, 1, 1]	0.4	0.4	0.914	2.625	0.400	2.638	2.62E-05	-0.49%
fig. 7.6 up-r	[1, 1, 1]	1	[1, 1, 1]	0.5	0.5	0.985	3.046	0.500	3.057	-2.45E-07	-0.35%
fig. 7.6 lo-le	[1, 1, 1]	1	[1, 1, 1]	0.6	0.6	1.046	3.440	0.599	3.456	1.02E-03	-0.45%
fig. 7.6 lo-mid	[1, 1, 1]	1	[1, 1, 1]	0.7	0.7	1.102	3.813	0.700	3.871	3.52E-04	-1.54%
	[1, 1, 1]	1	[1, 1, 1]	0.8	0.8	1.152	4.168	0.798	4.362	2.03E-03	-4.66%
fig. 7.6 lo-r	[1, 1, 1]	1	[1, 1, 1]	0.9	0.9	1.198	4.508	0.893	4.840	8.28E-03	-7.36%
	[1, 1, 1]	2	[2, 1, 1]	0.05	0.05	0.457	0.656	0.050	0.670	2.88E-04	-2.12%
	[1, 1, 1]	2	[2, 1, 1]	0.1	0.1	0.576	1.042	0.100	1.058	-7.64E-07	-1.54%
	[1, 1, 1]	2	[2, 1, 1]	0.15	0.15	0.659	1.365	0.150	1.374	2.77E-03	-0.67%
	[1, 1, 1]	2	[2, 1, 1]	0.2	0.2	0.726	1.654	0.200	1.669	-1.52E-09	-0.90%
	[1, 1, 1]	2	[2, 1, 1]	0.25	0.25	0.782	1.919	0.250	1.930	-7.25E-09	-0.55%
	[1, 1, 1]	2	[2, 1, 1]	0.3	0.3	0.831	2.167	0.300	2.180	-1.54E-06	-0.61%
	[1, 1, 1]	2	[2, 1, 1]	0.35	0.35	0.874	2.402	0.350	2.415	-2.25E-09	-0.54%
	[1, 1, 1]	2	[2, 1, 1]	0.4	0.4	0.914	2.625	0.400	2.668	5.95E-04	-1.61%
	[1, 1, 1]	2	[2, 1, 1]	0.45	0.45	0.951	2.840	0.450	2.853	6.95E-07	-0.48%
	[1, 1, 1]	2	[2, 1, 1]	0.5	0.5	0.985	3.046	0.500	3.063	4.08E-07	-0.53%
	[2, 1, 1]	2	[2, 1, 1]	0.05	0.1	0.576	1.042	0.100	1.054	4.78E-04	-1.13%
	[2, 1, 1]	2	[2, 1, 1]	0.1	0.2	0.726	1.654	0.200	1.667	1.69E-03	-0.78%
	[2, 1, 1]	2	[2, 1, 1]	0.15	0.3	0.831	2.167	0.300	2.178	6.99E-04	-0.48%
	[2, 1, 1]	2	[2, 1, 1]	0.2	0.4	0.914	2.625	0.399	2.660	2.35E-03	-1.33%
	[2, 1, 1]	2	[2, 1, 1]	0.25	0.5	0.985	3.046	0.500	3.060	1.21E-05	-0.44%
	[2, 1, 1]	2	[2, 1, 1]	0.3	0.6	1.046	3.440	0.600	3.455	2.69E-04	-0.43%
	[2, 1, 1]	2	[2, 1, 1]	0.35	0.7	1.102	3.813	0.699	3.841	1.19E-03	-0.76%
	[2, 1, 1]	2	[2, 1, 1]	0.4	0.8	1.152	4.168	0.799	4.220	1.73E-03	-1.25%
	[2, 1, 1]	2	[2, 1, 1]	0.45	0.9	1.198	4.508	0.899	4.602	1.57E-03	-2.08%
	[2, 1, 1]	2	[2, 1, 1]	0.5	1	1.241	4.836	0.998	4.980	1.69E-03	-2.98%
fig. 7.10	[3, 1, 1]	3	[3, 1, 1]	[0.3 0.1 0.3]	0.75	0.895	5.030	0.750	5.050	2.70E-04	-0.46%

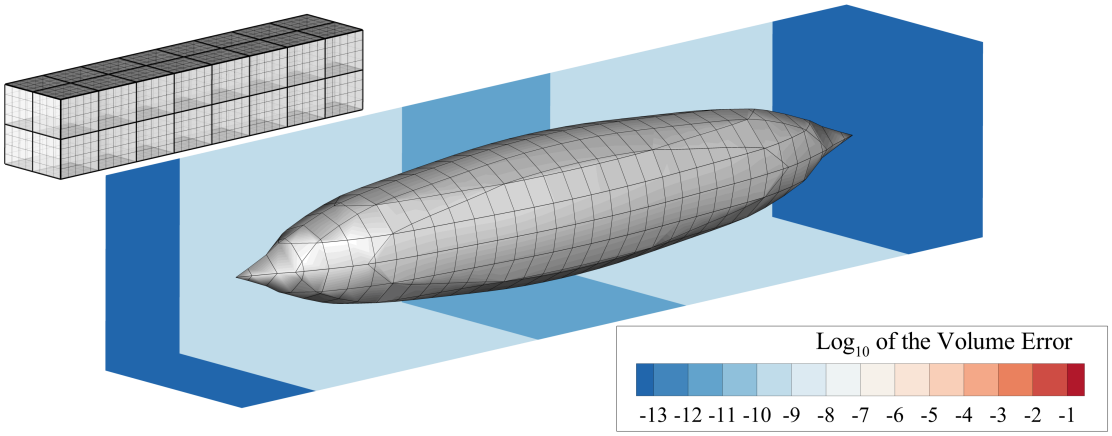


Figure 7.7: Sears-Haack body represented using 40 VOS cells in [10, 2, 2] layout. The colour in the colours in the background present the level of convergence of the r-surface on the correct volume.

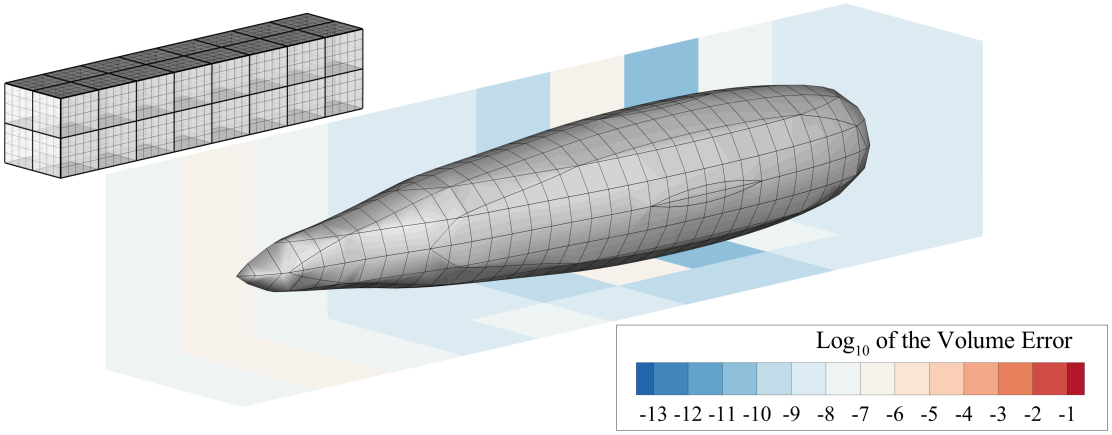


Figure 7.8: Truncated Sears-Haack body represented using 40 VOS cells in [10, 2, 2] layout. The colour in the colours in the background present the level of convergence of the r-surface on the correct volume.

sation community [261, 262]. Figure 7.9 presents a coarse representation of a wing using a [2,5,6] layout of design variables. This provides 10 volume fraction values to design the cross-section of the wing at six span locations. One of the side effects of building surfaces of minimum area is that long and slender profiles are not initially possible. To allow elongated bodies, the longer dimension of the surface needs to be de-weighted in terms of area. This can be achieved either inside the shape generation by multiplying the coordinates by individual weights or by externally altering the aspect ratio of the design grid.

Three dimensional optimisation usually relies on deformative methods starting from

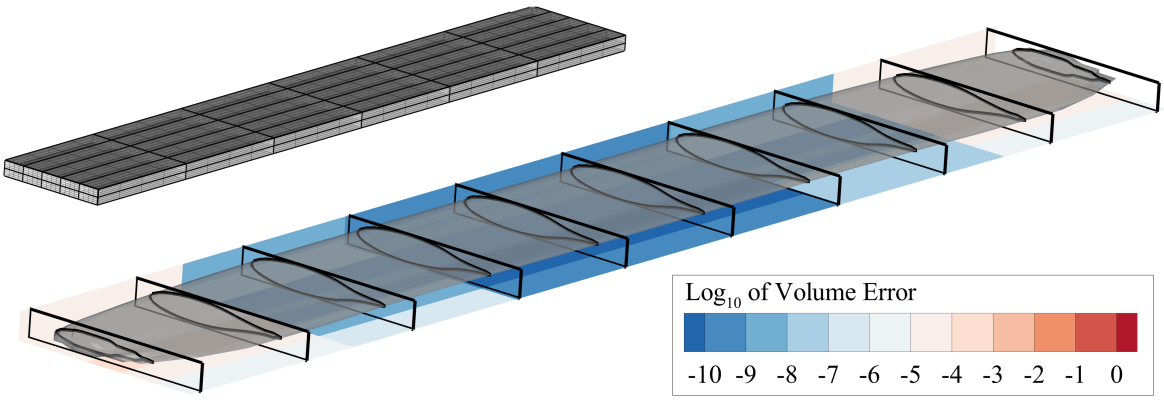


Figure 7.9: Coarse wing represented using 60 VOS cells in [2, 5, 6] layout. The colour in the colours in the background present the level of convergence of the r-surface on the correct volume.

a high quality discretisation. This presents a challenge in evaluating the RSVS: as a constructive parameterisation method it cannot easily benefit from an existing geometry. Three approaches are envisaged to resolve this issue: progressive design space refinement; the integrated parameterisation approach which was pioneered in Section 6.4; or, partial design space representations. Of these, hierarchical design variables have been shown to be effective in three dimensions to optimise aerodynamic features at a range of geometric scales concurrently [176, 263]. Despite these possibilities, design of an entire wing is not the primary use case of the RSVS: the RSVS will be targeted at cases where its topological flexibility, is an asset not a drawback.

7.2.4 Topological Flexibility of the 3 Dimensional RSVS

The minimal case to show the topological behaviour of the 3D-RSVS requires 3 VOS cells. Figure 7.10 presents the geometries generated by varying the value of the central VOS cell. Between values of 0.3 and 0.1 the topology of the geometry changes from a single body to 2 spherical bodies. The case generating two spheres is added to Table 7.1 and shows a similar geometric convergence on spheres as the cases discussed in the previous Section (7.2.2).

Figure 7.11 shows 4 different surfaces generated by the 3 dimensional RSVS. On the left the volume grid on which the volume fractions are specified (thick lines) and the snaking grid (thin lines) on which the restricted surface evolves. These surfaces illustrate some of the more complex topologies that can be achieved with a small set of design variables. While these topologies may not be of interest for external aerodynamic

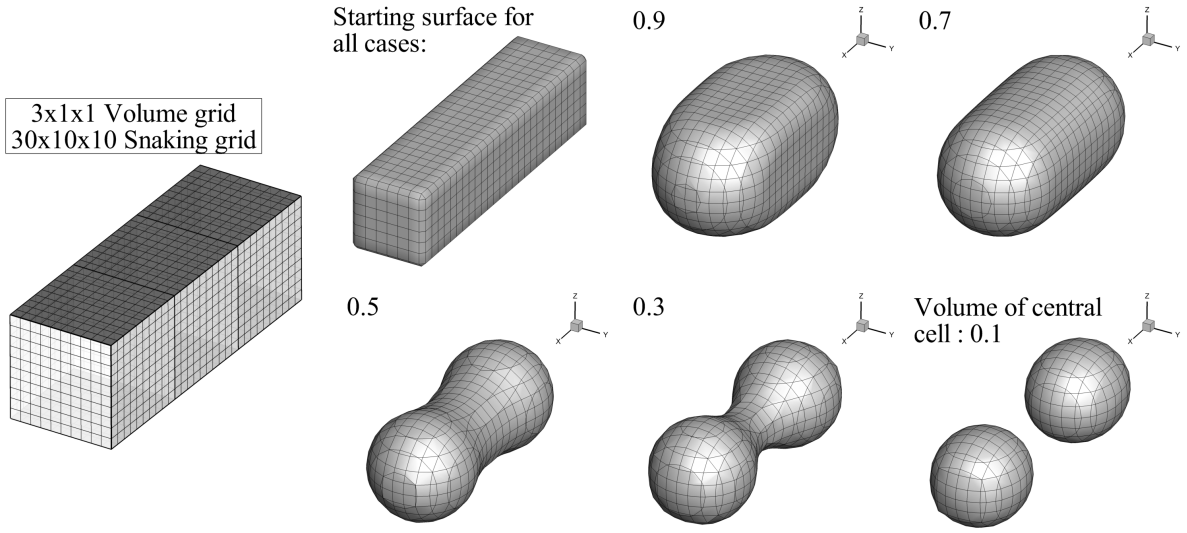


Figure 7.10: 5 different final geometries defined by 3 volume cells. The VOS at each end are kept constant while the volume fraction of the central cell is varied from 0.1 to 0.9.

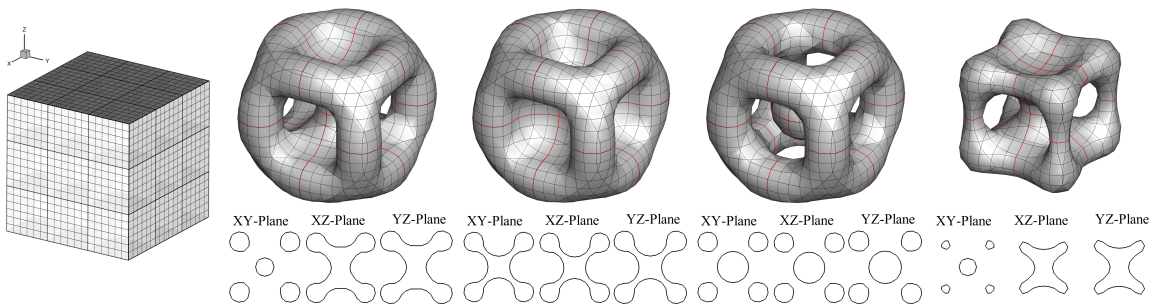


Figure 7.11: Four different final topologies defined by 27 VOS cells in a 3^3 layout (on the left). These cases illustrate the topological and smooth shape control 3D-RSVS provide with few design variables. To aid understanding of the topologies being represented slices through the centre of each dimension are provided.

optimisation, these could have application in the design of pipes or structures.

7.3 Implementation of the 3D-RSVS

This section delves into some of the detail of the 3D implementation motivating choices and exploring properties of the 3D-RSVS system. In particular the novel r-surface method is described and its implementation is outlined; finally the integration with the area minimisation defining the 3D-RSVS is presented. The implementation of the 3D-RSVS is done in C++ and is made available on GitHub [264] for contribution and download under the GNU Lesser-GPL license.

7.3.1 Rules for the Evolution of Water-Tight Surfaces

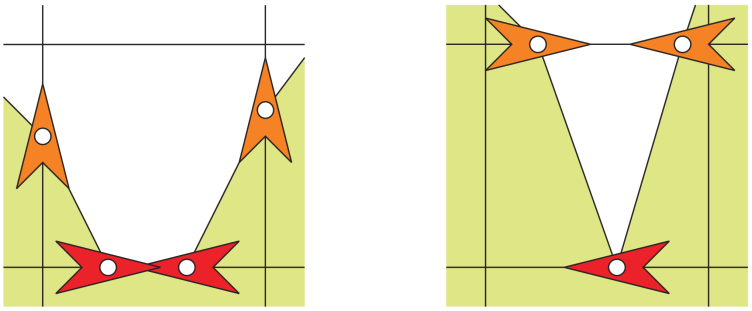
At the core of a three dimensional shape and topology parameterisation method must be an efficient method for topology evolution, compatible with the smooth and compact support requirements of aerodynamic parameterisation. The explicit evaluation of intersections between discrete or analytical geometries is a difficult and expensive problem. The r-surface simplifies those calculations by constraining control vertices to a grid, forcing intersections to happen point-to-point along edges. This property is very desirable in three dimensions as it reduces the cost of computing intersections to a search through a hashed map and, if necessary, a floating point comparison.

The extension of the restricted snake to surface objects relies on reformulating the two dimensional connectivity rules into a generalisable form. Recall that these rules, as described in [231], are:

1. No two connected snaxels can be on the same edge;
2. Snaxels must travel out of the profile.

In terms of connectivity the second rule manifests itself as: a snaxel cannot be connected to two edges which are part of the same face of the snaking grid. Examples of invalid connections are shown in Figure 7.12.

The initial connectivity rules can be formalised in terms of the relationship between r-snake elements (vertices and edges) and the underlying snaking grid elements (vertices, edges and faces). This observation enables the systematic extension of the connectivity rules to three dimensions. If two connected r-snake edges cannot be part of the same face of the snaking grid, it follows that two faces of a restricted surface cannot be part of the same volume cell of the underlying snaking grid.



(a) The snaxels in red are connected and on the same edge.

(b) The snaxel in red travels tangentially to the profile.

Figure 7.12: Invalid snaxel connections, image from [217].

The rules for this restricted surface (r-surface) are formalised into the following:

1. No two snaxels connected (by an r-surface edge) can be on the same *snaking grid* edge;
2. No two r-surface edges connected (by a snaxel) can be in the same *snaking grid* face;
3. No two r-surface faces connected (by r-surface edge) can be in the same *snaking grid* cell.

The 3-dimensional rules stated above can be further generalised to handle the marching of a N-dimensional restricted-polytope, including support for topology change. In all dimensions, there is a single special case for vertices which are 0-dimensional objects and all other rules are the same relative to the dimensionality of the object being handled. The N-dimensional rules are the following:

1. No two restricted-polytope 0-dimensional object connected (by a restricted-polytope 1-dimensional object) can be on the same *snaking grid* 1-dimensional object;
2. No two restricted-polytope z-dimensional object connected (by a restricted-polytope (z-1)-dimensional object) can be in the same *snaking grid* (z+1)-dimensional object for $z \in \{1, \cdots, N\}$.

The 3-dimensional rules were shown to work robustly and efficiently for arbitrarily complex geometries evolved on tetrahedral and hexahedral snaking-grids; the implementation is expected to work for all convex snaking grids (no internal angle above π). Algorithms, data structures and pseudo code for the current implementation of the r-surface are available in Section 7.3.2. Figures 7.13, 7.14 and 7.15 show the evolution of

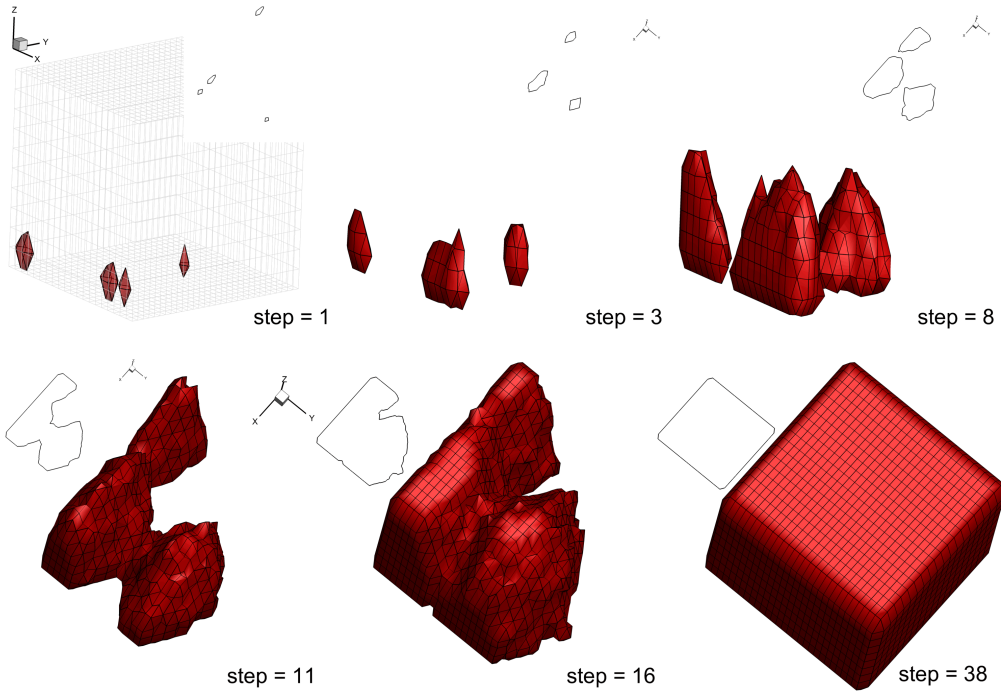


Figure 7.13: A restricted surface in a Cartesian snaking grid is evolved under a uniform velocity field. The surface is in red, the black outlines are a section through the surface at $z = 0.15$.

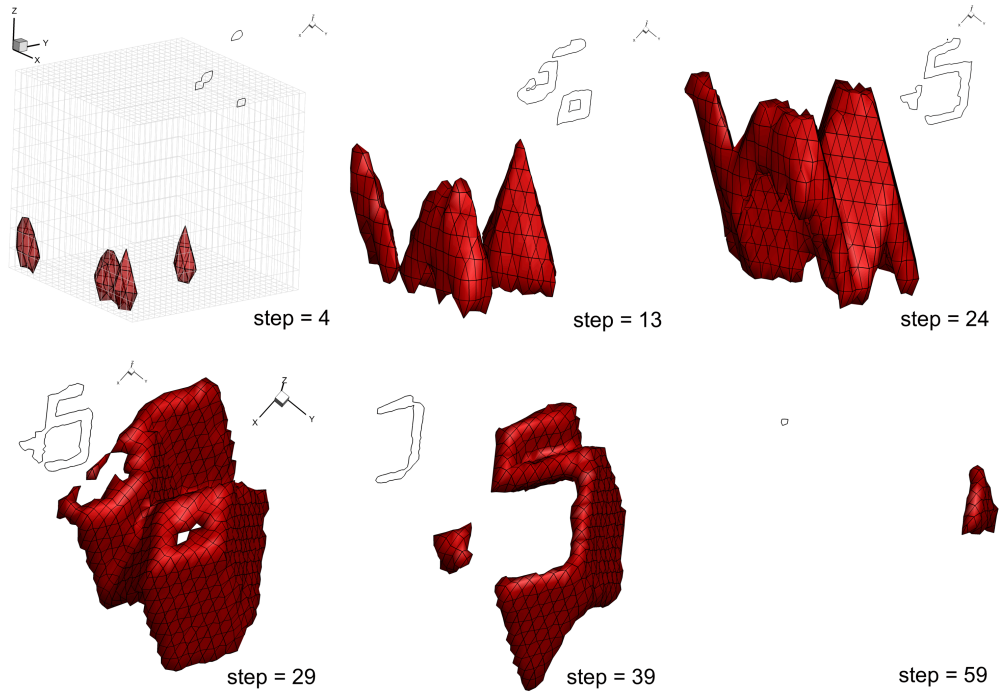


Figure 7.14: A restricted surface in a Cartesian snaking grid is evolved under a uniform velocity field with reflections at the design space boundary. The surface is in red, the black outlines are a section through the surface at $z = 0.15$.

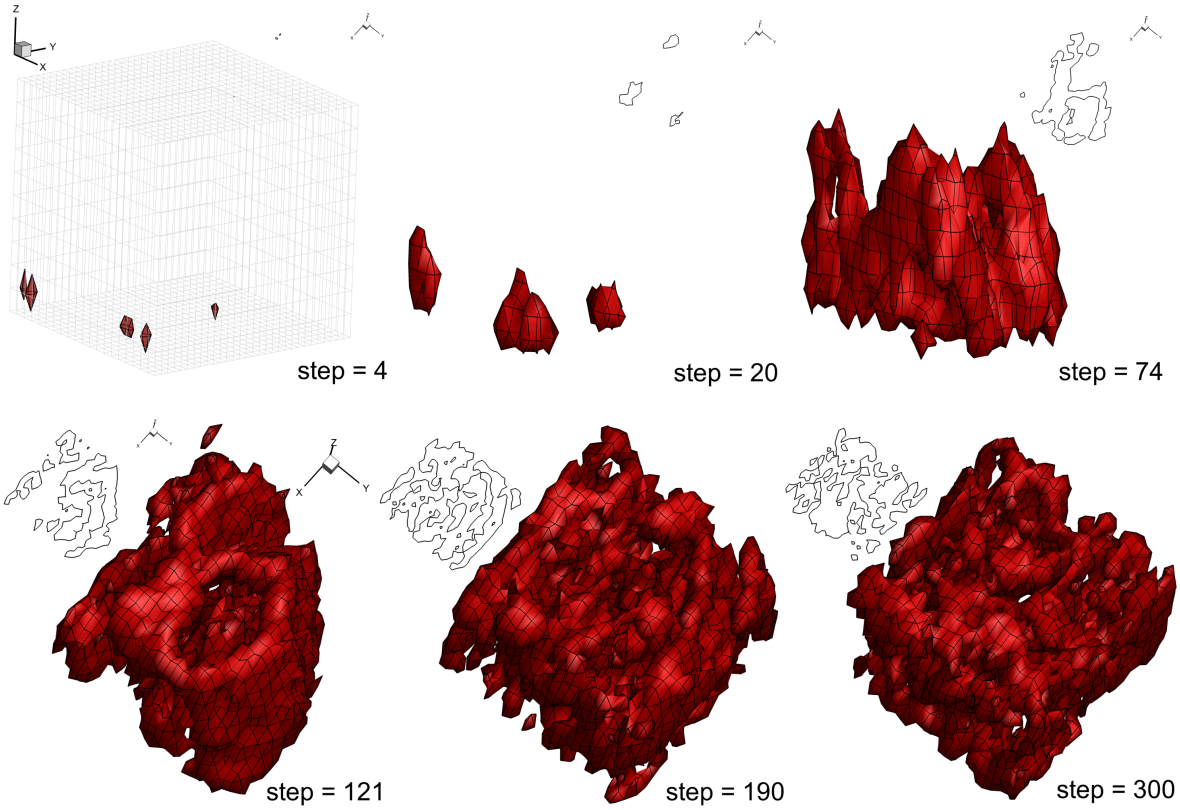


Figure 7.15: Test of the restricted surface process and rules: a restricted surface in a Cartesian snaking grid is evolved under a velocity field with random variations. The surface is in red, the black outlines are a section through the surface. The proposed restricted surface algorithm is robust and fast for arbitrary complex topologies.

r-surfaces spawned from 6 distinct vertices. Between these figures, only the algorithm for snaxel velocity update differs, respectively: unit velocity, unit velocity with reflection at the boundary and random velocity with reflection. This was performed as a test of the topological flexibility of the r-surface process, validating its use to evolve the 3D-RSVS surface.

The r-surface relies exclusively on connectivity information to detect collisions avoiding the need for expensive, floating point, intersection calculations. This process is efficient, robust and scalable: in Figure 7.15 the evolution features up to a thousand topological changes per step, processed in around a second per step.

7.3.2 Algorithms for the Implementation of the R-Surface

While the rules defining a valid r-surface have been stated in Section 7.3 they must now be implemented into a method for surface evolution. The restricted surface method relies on the data structures presented in Figure 7.16 and three core algorithms:

- **clean** the r-surface connectivity to have only valid elements (Algorithm 7);
- **cut and merge** geometries to evolve topology (Algorithm 9);
- **spawn** at snaking grid vertices (Algorithm 11).

The removal of invalid connections in the restricted surface relies on the combination of grid elements, and, the removal of orphaned elements. These two steps applied in the correct order for vertices, edges, faces and volume cells allows invalid connections to be pruned; Algorithm 11 presents the steps necessary to clean up those connections. In addition to these core processes two composited algorithms are used to complete the r-surface method:

- **snake initialisation** to generate the starting restricted surface (Algorithm 12);
- **grid vertex crossing** to enable the progression of the r-surface through the snaking grid (Algorithm 10).

Algorithm 7 Restricted surface algorithm for spawning at a snaking grid vertex.

To spawn at a vertex vx_i
for Each snaking grid edge e_j in vx_i`.edgeind` **do**
    Add a new snaxel on edge e_i
end for
for Each snaking grid face f_j in vx_i`.edgeind[].surfind` **do**
    Add a new edge in face f_j
    Connect this new edge to snaxels on edges f_j`.edgeind`
end for
for Each snaking grid cell c_j in ve_i`.edgeind[].surfind[].voluind` **do**
    Add a new face in volume element c_j
    Connect this new face to edges in faces c_j`.surface`
end for
Connect all faces with a new volume element.

`int`	Index			
`double`	VOS	Target VOS	VOS error	Volume
`vector<int>`	Surface indices (`surfind`)			

(a) Data structure of a restricted surface volume element ($d = 3$).

`int`	Index	Snaking grid volume index (`parentind`)		
`double` (for 2D)	VOS	Target VOS	VOS error	Area
`vector<int>`	Edge indices (`edgeind`)		Volume indices (`voluind[2]`)	

(b) Data structure of a restricted surface face element ($d = 2$).

`int`	Index	Snaking grid face index (`parentind`)	
`double`	Length		
`vector<int>`	Vertex indices (`vertind[2]`)	Surface indices (`surfind`)	

(c) Data structure of a restricted surface edge ($d = 1$).

`int`	Index	Snaking grid edge index (`parentind`)
	Origin vertex (`fromvert`)	Destination vertex (`tovert`)
	Frozen status (`isfreeze`)	
`double`	Normalised distance (`d`)	Normalised speed (`v`)
`vector<double>`	Length `l[3]`	
`vector<int>`	Edge indices (`edgeind`)	

(d) R-surface vertex (snaxel) element data structure ($d = 0$).

Vector of volume objects	Vector of face objects
Vector of edge objects	Vector of snaxel objects
`vector<bool>` are snaking grid vertices inside the r-surface?	

(e) R-surface data structure.

Figure 7.16: Definitions of the data structure used in the restricted surface process.

Algorithm 8 3D-RSVS element combination algorithm

if Elements (e_1 and e_2) of dimension n are identified for merging **then**
  Replace e_2 by e_1 in all connectivity lists.
  Mark e_2 for deletion.
end if

Algorithm 9 Restricted surface algorithm for cutting/merging of bodies.

for All snaxels s_i and s_j carried by the same grid edge **do**
  if s_i and s_j are in the same location **and** are not moving apart **then**
    Mark snaxels s_i and s_j for combination.
  end if
end for
Apply the snaxel combination. This creates invalid connections.
Run the connectivity clean-up of Algorithm 11.

Algorithm 10 Restricted surface algorithm for crossing through snaking grid vertices.

for All snaxels s_i **do**
  if s_i.d== 1 **and** s_i.v> 0 **then**
    Mark snaking grid vertex s_i.tovert for spawning.
  else if s_i.d== 0 **and** s_i.v< 0 **then**
    Mark snaking grid vertex s_i.fromvert for spawning.
  end if
end for
for Each snaking grid vertex vx_i marked for spawning because of snaxel(s) $s_{i1,\cdots,im}$. **do**
  Apply the spawning algorithm (Algorithm 7) at vertex vx_i
end for
Apply the algorithm for topology cutting/merging (Algorithm 9) to all the new snaxels.

Run the connectivity clean-up of Algorithm 11.

Algorithm 11 Restricted surface connectivity clean-up process (notation defined in Figure 7.16).

while Changes of connectivity are detected **do**
  for All snaxels s_i **do**
    if Snaxel s_i is disconnected from the r-surface **then**
      Mark snaxel s_i for deletion
    end if
  end for
  Remove snaxels marked for deletion.
  for All r-surface edge e_i **do**
    if Edge is connected to a single vertex: e_i.`vertind`[1] == e_i.`vertind`[2] **then**
      Mark edge e_i for deletion.
    else if e_i.`vertind`[1].`parentind`== e_i.`vertind`[2].`parentind` **then**
      Mark snaxels e_i.`vertind` 1 and 2 for combination.
      Mark edge e_i for deletion.
    end if
  end for
  Apply snaxel combination operations (Algorithm 8).
  Remove edges marked for deletion.
  for All r-surface edge e_i **do**
    if e_i.`vertind`[l] == e_j.`vertind`[m] **and** e_i.`parentind`== e_j.`parentind` **then**
      Mark edges e_i and e_j for combination.
    end if
  end for
  Apply snaxel combination operations.
  for All r-surface face f_i **do**
    if f_i.`edgeind`[l] == f_j.`edgeind`[m] **and** f_i.`parentind`== f_j.`parentind` **then**
      Mark faces f_i and f_j for combination.
    end if
  end for
  Apply face combination operations.
end while

Algorithm 12 Restricted surface algorithm for initialisation at the boundary of the void domain.

1: *Note: identify vertices around which lie between region of empty VOS and non-empty VOS.*
2: **for** All faces f_i in the snaking grid **and** the VOS grid **do**
3: **if** f_i`.voluind[0].target==` 0 **and** f_i`.voluind[1].target>` 0 **then**
4: Mark all vertices connected to this face for spawning f_i`.edgeind[].vertind`
5: **end if**
6: **end for**
7: **for** Each snaking grid vertex vx_i marked for spawning because of snaxel(s) $s_{i1,\cdots,im}$. **do**
8: Apply the spawning algorithm (Algorithm 7) at vertex vx_i
9: **end for**
10: Take a step of length $d = 0.5$ for all snaxels.
11: Run the topology cutting/merging Algorithm 9.
12: *Note: At this stage two surfaces exist for each block of faces: one just outside the boundary and one just inside.*
13: remove the outside surface, identifying it using a flood fill on vertices lying on edges which are in cells with a VOS target of 0.
14: **for** Each face f_i of the r-surface **do**
15: **if** f_i`.parentind.target(VOS)==` 0 **then**
16: Mark f_i for removal.
17: Use flood fill to mark all faces, edges and vertices of the r-surface connected to f_i for deletion.
18: **end if**
19: **end for**
20: Process requested element removals.
21: Reverse surface direction by flipping snaxels (`s.fromvert`↔`s.tovert` and `s.d=`1−`s.d`)
22: Identify snaking grid vertices inside the r-surface using flooding from snaxels origin vertex.
23: Build r-surface volumes by identifying blocks of internal grid vertices and connected surfaces.

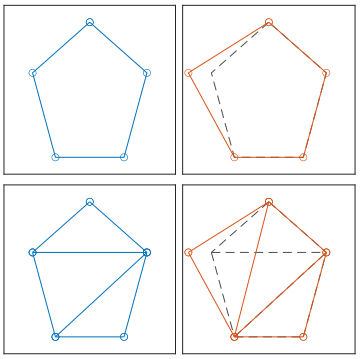
7.3.3 Triangulation of Restricted Surfaces into Polyhedra

In order to compute the area and volume of a geometry and drive the 3D-RSVS process an expression for the shape of those faces is required. While the rules for building the r-surface guarantee the formation of water-tight surfaces, it does not guarantee that the surface will be a polyhedron: some of the faces may not be flat. This is because the r-surface is controlled by the positioning of its vertices with the rest of the geometry derived from the connectivity information forced by the snaking grid. Since an analytical solution to the 3D-RSVS formulation has not been found, another approach to define the position of the boundary is needed.

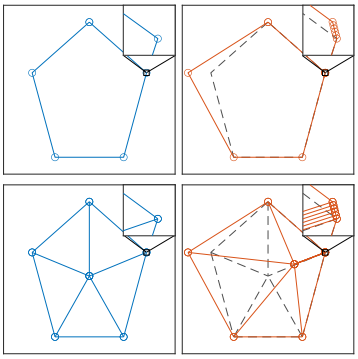
The most natural approach to arrive at an explicit boundary is to triangulate the restricted surface. A triangulation guarantees flat faces, meaning that the output of the process is always a valid polyhedron; in turn this allows reliable calculation of the volume and area enclosed by the surface. By triangulating all faces with more than 3 edges, regardless of whether they were flat or not, simplifies the implementation of the volume and area calculations and differentiations as only triangular faces will need to be processed. In order to be effective in the r-surface context the triangulation needs to guarantee a smooth response of the area and the volume of the polyhedron; the following observations are made:

1. the triangulation connectivity must not change when the snaxels move, or in other words, triangulation connectivity must depend only on r-surface connectivity, this rule is breached by Figure 7.17a;
2. the triangulation properties (area, volume) must only depend on the geometric properties of the surface, not on its connectivity, this rule is breached by Figure 7.17b.

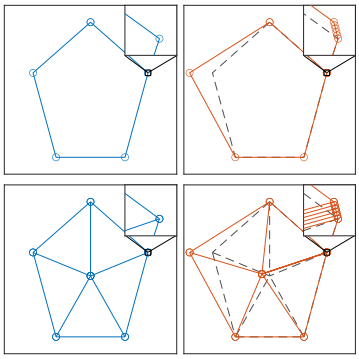
In light of these observations, three triangulations are considered in Figure 7.17: a Delaunay triangulation (Figure 7.17a), a triangulation built by linking every vertex of the polygon to the average point of those vertices (Figure 7.17b); and, one built by linking every vertex to the centroid of the curve defining the polygon (Figure 7.17c). The triangulation relying on the mean position of the vertices fails when a r-surface crosses through a snaking grid vertex: at that point, multiple vertices will lie very close to each other, unduly impacting the triangulation. To achieve consistent connectivity and the smooth response through changes of r-surface connectivity the triangulation of faces is built around point $\bar{\mathbf{c}}$ which is the mean position of face vertices weighted by edge length. Equation 7.3 formalises this process for a closed face with $n+1$ vertices and the last vertex repeated.



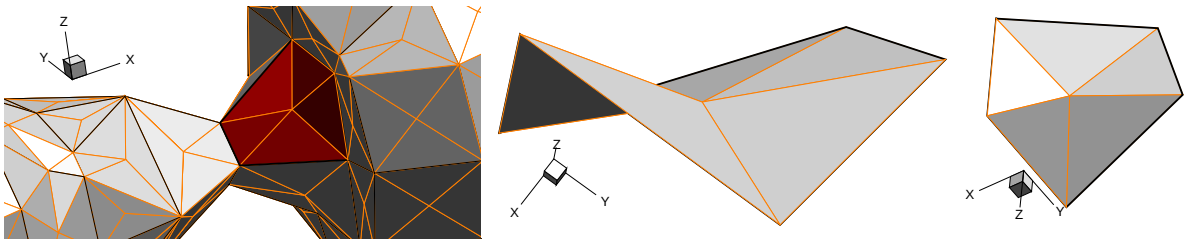
(a) Delaunay triangulation



(b) Triangulation through the mean of the control points



(c) Triangulation through the centroid of the curve (Equation 7.3)



(d) Triangulation through the centroid for a non-flat face.

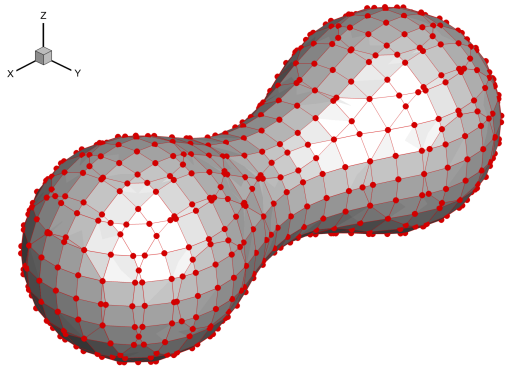
Figure 7.17: Triangulation of a pentagon and a pentagon with one moved vertex, and one repeated vertex with three possible triangulations. Only the contour length weighted centroid (Figure 7.17c), fulfils the stability requirements of the 3D-RSVS parameterisation.

$$\bar{\mathbf{c}} = \frac{\sum_{i=1}^{n} \|\mathbf{p}_{i+1} - \mathbf{p}_i\| (\mathbf{p}_{i+1} + \mathbf{p}_i)}{2 \sum_{i=1}^{n} \|\mathbf{p}_{i+1} - \mathbf{p}_i\|} \qquad \text{with} : \mathbf{p}_{n+1} = \mathbf{p}_1 \tag{7.3}$$

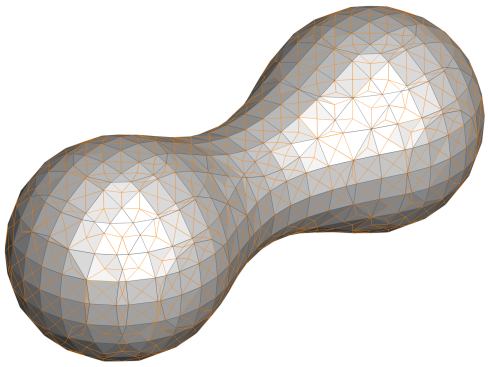
The r-surface face is triangulated by linking this pseudo-centroid of the face to each of the vertices. This formulation is used as it prevents changes in connectivity, due to movements of the restricted surface, to cause jumps in the position of $\bar{\mathbf{c}}$; it is not affected by duplicate points. Figure 7.17 shows the stability of the centroid calculation with changes of connectivity, allowing smooth evolution of the surface necessary for the r-surface method to converge on a solution of the 3D-RSVS governing equation.

7.3.4 Evaluation of the 3D-RSVS Equations on the R-Surface

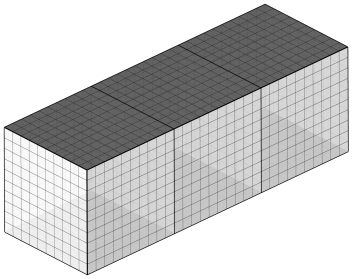
To drive the position of the restricted surface the original continuous area minimisation problem (Equation 7.1) needs to be discretised in terms of the r-surface snaxel variables.



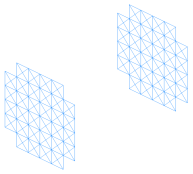
(a) Restricted surface



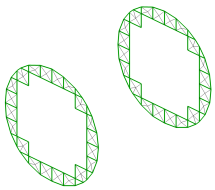
(b) Triangulated restricted surface



(c) VOS and snaking domains



(d) Triangulated VOS cell faces active in the volume calculations



(e) Triangulation of VOS cell faces cut by the r-surface

Figure 7.18: Triangulations necessary for the calculation of the objective function and constraints of the 3D-RSVS problem.

This process is achieved by expressing the area and volume integrals on the triangulated surface; a discrete form of the governing mathematical program can then be defined.

The discrete variables used for the 3D-RSVS are very similar to the two dimensional implementation: the same six properties from the r-surface geometry and the snaxel positions are needed. These properties are:

- the snaxel index (i), used to reference it in all operations;
- the normalised position along an edge ($d_i \in [0,1]$);
- the scalar velocity along that edge ($v_i \in \mathbb{R}$);
- the snaxel position in Cartesian coordinates ($\mathbf{p}_i$);
- the direction of travel of the snaxel ($\Delta\mathbf{g}_i$) and the vertex of origin ($\mathbf{g}_{\mathbf{i},\mathbf{1}}$);
- the normal vectors to the preceding and following edges ($\mathbf{n}_i$ and $\mathbf{n}_{i+1}$).

These properties were represented graphically in Figure 7.5.

The continuous expressions presented in Section 7.1.1 are easily computed for polyhedra with triangular faces. The following notation is adopted: S refers to the r-surface;

C to cells of the VOS grid. The area of each triangular face (A_{S_k}) is computed using Equation 7.4, only the position of each of the corner vertices is required ($\mathbf{p_0}, \mathbf{p_1}, \mathbf{p_2}$). This approach is also applicable to $V_{S,k}$, the contribution of face k, to the volume of the polyhedron. The volume contributions from the underlying grid ($V_{C_j,k}$) are also taken into account. Vertices represented by symbol $\mathbf{p}_{i,k}$ are *active* vertices (snaxels or pseudo-centroids) which move with the surface being designed, *static* vertices which are part of underlying grids are represented by symbol $\mathbf{g}_{i,k}$.

$$A_{S,k} = \frac{1}{2} \left\| \left(\mathbf{p}_{1,k} - \mathbf{p}_{0,k}\right) \times \left(\mathbf{p}_{2,k} - \mathbf{p}_{0,k}\right) \right\| \tag{7.4}$$

$$V_{S,k} = \frac{1}{6} \left(\mathbf{p}_{0,k} \cdot \left(\left(\mathbf{p}_{1,k} - \mathbf{p}_{0,k}\right) \times \left(\mathbf{p}_{2,k} - \mathbf{p}_{0,k}\right) \right) \right) \tag{7.5}$$

$$V_{C_j,k} = \frac{1}{6} \left(\mathbf{g}_{0,k} \cdot \left(\left(\mathbf{g}_{1,k} - \mathbf{g}_{0,k}\right) \times \left(\mathbf{g}_{2,k} - \mathbf{g}_{0,k}\right) \right) \right) \tag{7.6}$$

The equations presented above can be assembled to calculate the volume of the polyhedron formed by the intersection of the r-surface and the faces of the *design grid*, this quantity is represented by value $V_{S\cap C,j}$. Components of this polyhedron are represented in Figure 7.18.

$$V_{S\cap C,j} = \frac{1}{6} \sum_{k=j_S(1)}^{j_S(q_{jS})} \mathbf{p}_{0,k} \cdot \left(\left(\mathbf{p}_{1,k} - \mathbf{p}_{0,k}\right) \times \left(\mathbf{p}_{2,k} - \mathbf{p}_{0,k}\right) \right) + \frac{1}{6} \sum_{k=j_C(1)}^{j_C(q_{jC})} \mathbf{g}_{0,k} \cdot \left(\left(\mathbf{g}_{1,k} - \mathbf{g}_{0,k}\right) \times \left(\mathbf{g}_{2,k} - \mathbf{g}_{0,k}\right) \right) \tag{7.7}$$

$j_S(\{1, \cdots, q_{jS}\})$ and $j_C(\{1, \cdots, q_{jC}\})$ are indexing functions specified for each design cell selecting the correct vertices respectively from the triangulated r-surface and the volume grid. The equations for volume and area of the polyhedra formed by the intersection of the r-surface and the design grid are derived and are now substituted into the mathematical program which defines the RSVS problem in 3-dimensions (Equation 7.1). This process leads to Equation 7.2 (repeated below) for the 3D-RSVS discretised by a triangulated r-surfaces.

$$\min_{\mathbf{d}} \sum_{k=1}^{q} A_{S,k}\left(\mathbf{p}_{0,k}, \mathbf{p}_{1,k}, \mathbf{p}_{2,k}\right) \quad \text{with } \mathbf{p}_{i,k}(\mathbf{d})$$

$$\text{s.t.} \sum_{k=j_S(1)}^{j_S(q_{jS})} V_{S,k}\left(\mathbf{p}_{0,k}, \mathbf{p}_{1,k}, \mathbf{p}_{2,k}\right) + \sum_{k=j_C(1)}^{j_C(q_{jC})} V_{C,k}\left(\mathbf{g}_{0,k}, \mathbf{g}_{1,k}, \mathbf{g}_{2,k}\right) = V_j \quad \forall j \in \{1, \cdots, m\}$$

Building a 3D-RSVS surface consists in finding the positions $\mathbf{d}$ of the r-surface snaxels which solve the mathematical program of Equation 7.2. This formulation has the benefit of being very general, it can be tackled on an arbitrary volume grid with any underlying snaking grid with any optimisation method.

7.3.5 Restricted-Surface Marching

As was the case for the 2D-RSVS the 3D implementation relies on SQP to solve the minimisation problem defining the 3D-RSVS surfaces. SQP provides convergence in a limited number of iterations for problems for which derivatives are available, which is the case for the discrete form of the 3D-RSVS problem presented in Equation 7.2.

The Newton step SQP equations presented below are derived in Boggs and Tolle [49]; only Equation 7.8 for the update of the snaxel velocities is shown here. The evaluation of the SQP (Equation 7.8) requires the following derivatives to be calculated: the Jacobian of the constraints ($\nabla_{\mathbf{d}}\mathbf{h}$); the gradient of the objective ($\nabla_{\mathbf{d}} f$) and the Hessian of the objective ($\mathbf{H}_{\mathbf{d}} f$).

$$\begin{aligned}
\boldsymbol{\lambda}^{k+1} &= \left((\nabla_{\mathbf{d}}\mathbf{h})^T (\mathbf{H}_{\mathbf{d}} f)^{-1} (\nabla_{\mathbf{d}} f)\right)^{-1} \left(\mathbf{h} - (\nabla_{\mathbf{d}}\mathbf{h})^T (\mathbf{H}_{\mathbf{d}} f)^{-1} (\nabla_{\mathbf{d}} f)\right) \\
\Delta_{\mathbf{d}}^{k+1} &= \mathbf{d}^{k+1} - \mathbf{d}^k = -(\mathbf{H}_{\mathbf{d}} f)^{-1} \left((\nabla_{\mathbf{d}} f) + (\nabla_{\mathbf{d}}\mathbf{h})\,\boldsymbol{\lambda}^{k+1}\right) \\
&\text{with}: f = A(\mathbf{d}) \quad \text{and } \mathbf{h} = \mathbf{V}(\mathbf{d})
\end{aligned} \tag{7.8}$$

Differentiation of f and $\mathbf{h}$ was done using the MATLAB Symbolic Toolbox, which allows analytical differentiation of the terms and the generation of the corresponding C code. This process greatly reduces the difficulty of differentiating the equations presented in Equations 7.4 and 7.7 and the risk of a mistake in the implementation. Because the differentiation is algorithmically developed from the area, volume and centroid functions; validation of the functions also validates the implementation of the derivatives. The C++ implementation also allows both analytical and finite difference gradients to be used; comparison showed them to be consistent.

This algorithmic process was applied to the differentiation of:

- $A_{S,k}$ with regard to $\mathbf{p}_{\mathbf{i},\mathbf{k}}$;
- $V_{C\cap S,k}$ with regard to $\mathbf{p}_{\mathbf{i},\mathbf{k}}$.
- $\bar{\mathbf{c}}$ with regard to $\mathbf{p}_{\mathbf{i},\mathbf{k}}$.

These differentiations are then expressed in terms of $\mathbf{d}$ using the matrix equation chain rule.

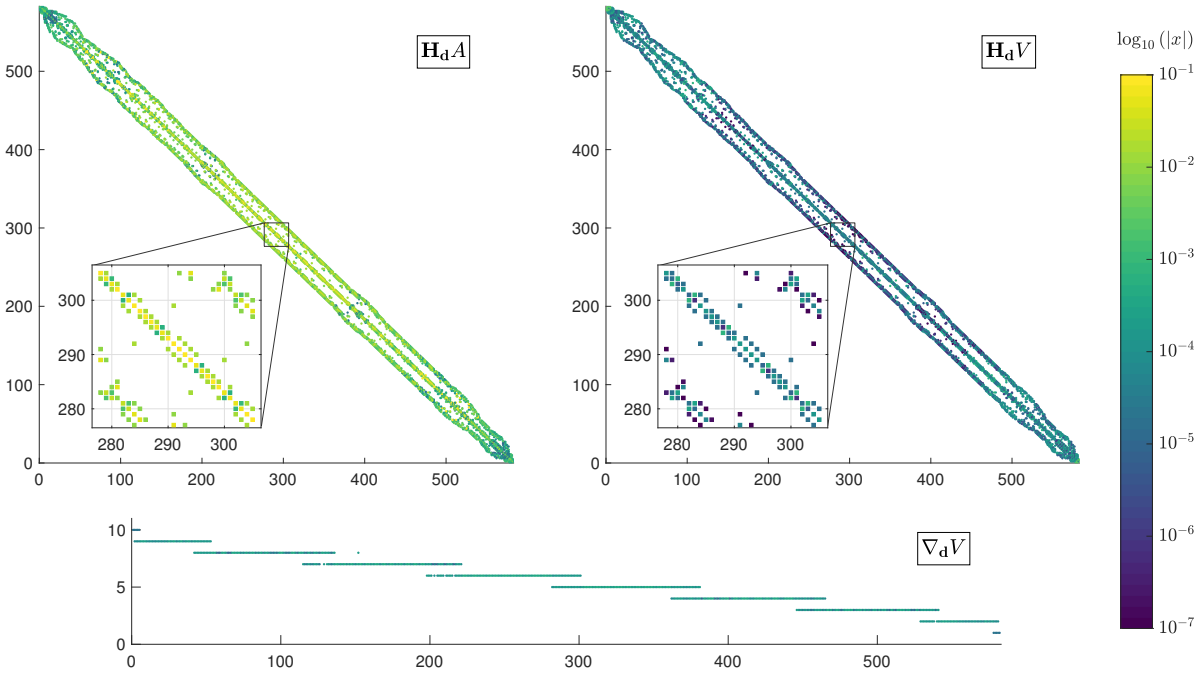


Figure 7.19: Sparsity patterns of the Hessian of the objective function and the Hessian and Jacobian of the constraints. These values are for the geometry of Figure 7.7.

While most properties of the RSVS carry over from the two to the three dimensional implementation, one notable difference is the expected size of the system. Whereas most 2D geometries can be accurately represented with a few hundreds to a few thousand control points, representation of complex topologies in 3D will usually take thousands to tens of thousands of snaxels. With large systems solving the SQP rapidly becomes expensive and methods are required to reduce the cost of the operations. For very large systems it may be necessary to switch to optimisation methods better suited to very large numbers of design variables, like Sequential Linear Quadratic Programming (SLQP), Sequential Linear Programming (SLP), or interior point methods.

The system that must be solved to compute the 3D-RSVS is sparse; this property can be exploited to speed up calculation for systems of intermediate sizes (thousands and low tens of thousands of points). Figure 7.19 shows the Hessian and Jacobian of the constraint and of the objective, for the profile in Figure 7.7. Sparsity is critical in reducing the memory footprint of the SQP calculation, letting it scale linearly instead of quadratically with the number of snaxels. The current implementation exploits both dense [265] and sparse [266] solvers using the Eigen library [267] for matrix mathematics in C++.

7.4 Properties of the 3D-RSVS Design Space

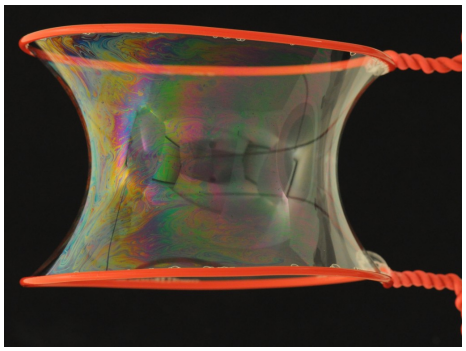
In this section, a comparison to minimal surfaces and an equivalence with surfaces of mean curvature are established. In addition, the wide range of possible design spaces enabled by the formulation of the 3D-RSVS is explored.

7.4.1 Discrete Differential Geometry as a Foundation of the 3D-RSVS

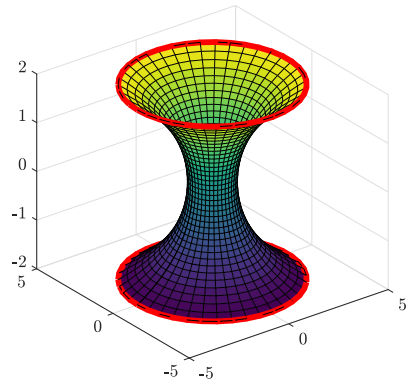
The study of minimal surfaces concerns itself with the discovery of surfaces of minimal area for a set of boundary conditions; these surfaces have a physical analogy: (open) soap films on wire frames. Provided that the air pressure is the same on both faces of the soap film, these naturally tend to minimise area. To develop solutions for a given boundary condition, the minimal surface problem can be formulated in a number of ways, one of which is calculus of variations. The first minimal surface to be described explicitly was the catenoid by Euler in 1741 (Figure 7.20) followed by the helicoid by Meusnier in 1776 (Figure 7.21) [260]. While steady progress had been made in the intervening time, the advent of computers in the 1980s along with the development of advanced mathematical machinery has allowed an explosion in the number and scope of discoveries in the field [260].

The relationship between the 3D-RSVS and minimal surfaces is obvious: both aim to minimise the area functional; however the 3D-RSVS differs in two ways: it is a free standing closed body, and it has a “pressure differential” between its faces in the form of the area constraint. In fact, minimal surfaces are part of a broader family of bodies knowns as surfaces of constant mean curvature (CMC); where minimal surfaces are a special case of CMCs for which the mean curvature is 0. The mean curvature is simply the sum of the principal curvatures. Continuing the “soap-film” analogy, surfaces of constant mean curvature are the result of minimising the area of a boundary between domains at different pressures: where minimal surfaces are films, CMCs allow bubbles. In general, the definition of a CMC surface is made in terms of a constrained volume [270–272], making the 3D-RSVS an obvious relative of these surfaces. Like minimal surfaces, a number of analytically described constant mean curvature surfaces exist: spheres, Wente tori [273], and the periodic P-CMC family [271]; however general explicit solutions for arbitrary boundary conditions do not exist.

Minimal surfaces and CMCs are found in nature as they allow the minimisation of internal stresses (films, bubbles and polymer mixtures), and optimise the assignment



(a) Soap film forming a catenoid from soapbubble.dk [268].



(b) Catenoid (boundary in red)

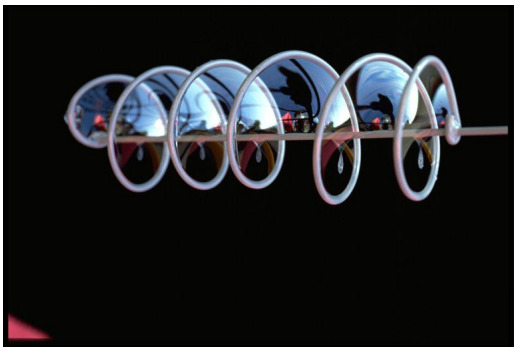
$$
\begin{aligned}
x &= c\cosh\left(\frac{v}{c}\right)\cos u \\
y &= c\cosh\left(\frac{v}{c}\right)\sin u \\
z &= v
\end{aligned}
\qquad
\begin{aligned}
&\text{where } u \in [-\pi, \pi) \\
&\text{and } v \in \mathbb{R} \\
&\text{and } c \text{ a constant}
\end{aligned}
$$

(c) Catenoid equation

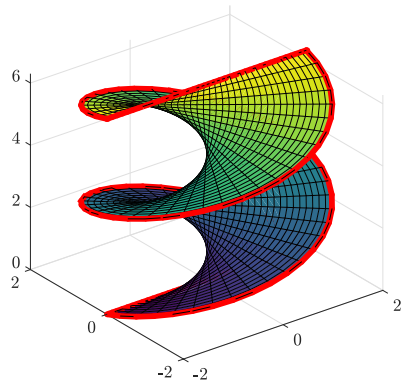
Figure 7.20: Soap films, equivalent minimal surfaces and explicit definitions for the catenoid.

of resources [260] (leaves of holly). These properties have motivated their use in architectural design and polymer micro structures. While analytical results for the 3D-RSVS may be out of reach at the moment, the progress in discrete differential geometry offers means of developing and classifying the results of the parameterisation. Of interest to the development of the 3D-RSVS are the numerical solvers for CMCs; notably the "Surface Evolver" by Brakke [274] has had a large impact on the field. This tool allows the evolution of a large number of surfaces, under a variety volume, boundary (wire-frames) and vertex constraints; with constant mesh topology. Later developments of these methods have led to mesh optimizing versions of the Surface Evolver which allowed topological cuts of the mesh, upon the collapse of faces [272].

From the previous discussion it appears that the 3D-RSVS has a very similar definition to CMC surfaces, with the difference that the RSVS resolves multiple, local, volume constraints rather than a single global one for CMCs. The geometry parameterised by the 3D-RSVS is in fact the solution to a set of simultaneous CMC problems, with the boundary conditions separating those CMC patches also unknown. Based on the



(a) Helicoidal soap bubble, from Exploratorium Teacher Institute [269]



(b) Helicoid (boundary in red)

$$\begin{aligned} x &= u\cos(cv) \\ y &= u\sin(cv) \\ z &= v \end{aligned} \qquad \begin{aligned} &\text{where } u,\ v \in \mathbb{R} \\ &\text{and } c \text{ a constant} \end{aligned}$$

(c) Helicoid equation

Figure 7.21: Soap films, equivalent minimal surfaces and explicit definitions for the helicoid.

similar formulations between the 3D-RSVS, and CMCs and the derivation of the two dimensional RSVS equivalence to non-uniform rational B-Splines (NURBS); it seems a reasonable hypothesis, that the continuous limit curve of the 3D-RSVS process are G_1 continuous CMCs patches.

This hypothesis is supported by Table 7.2 which showed the tendency of the 3D-RSVS to produce spheres for simple layouts of design variables. While an empirical or analytical proof of this hypothesis is not provided in this thesis, a path to one is suggested.

1. Solve the 3D-RSVS problem for a set of VOS design variables;
2. for each VOS cells extract the boundaries generated by the intersection with the r-surface;
3. for each boundary generate a CMC patch using “Surface Evolver” [274] or the PVT-CMC of Pan et al. [272];
4. compare the surfaces to the RSVS, if the hypothesis is true, the solutions should converge as the discretisations converge.

It was not followed as this potential relationship was only discovered late in the redaction

of this document.

The similarity between the 3D-RSVS problem and CMC surfaces has broader implications for the development of the implementation. The CMC and RSVS systems have similar properties; this suggests that methods used in CMC solvers to guarantee convergence [275] and smooth meshes [272] may be used to alleviate the current limitations of the 3D-RSVS. Another implementation of CMC surfaces is available as a plug-in [276] to the computational aided design (CAD) package *Rhinoceros 3D* [277]. These tools for the calculation of discrete CMCs are all interactive, meaning that the current cost of the 3D-RSVS is not intrinsic to the method.

7.4.2 Tailoring of Volume of Solid Grids

The integration of topology optimisation tools and CMC surface engines into CAD distributions reveals that the scope of the 3D-RSVS may be broader than shape and topology parameterisation for aerodynamics. In fact, the 3D-RSVS is a generic design tool with close links to structural topology optimisation (STO) density methods, level set methods, NURBS, and, CMC surfaces. These relationships suggest that the RSVS could be used interactively as a prototyping tool which natively supports compact and smooth topology optimisation. While the need for compactness in many optimisation applications has been reduced by the ubiquity of adjoint solvers, it is necessary that design modes be intuitive to designers.

While Cartesian VOS layouts are an effective test of the initial implementation of the parameterisation, these grids do not exploit the somewhat unique flexibility of the 3D-RSVS to generate smooth shapes out of arbitrary design spaces. The use of non-Cartesian volume spaces, has two benefits: the number of unused design variables for a given resolution can be significantly reduced, and, modal responses can be tailored to a geometry. The following four grid modifications are envisaged:

1. **global grid deformations:** rotations, shears and stretches applied to the grid to replicate usual engineering design modes (e.g. in aerodynamics span, sweep, thickness);
2. **local grid deformations:** would likely behave in a similar way to an FFD control cage on the design;
3. **voxel based refinement:** extension of the 2D RSVS refinement process;
4. **Voronoi design cells:** user defined points form a design space by using their Voronoi diagram as the VOS cells of the 3D-RSVS, this process allows extreme flexibility and relatively easy user interface.

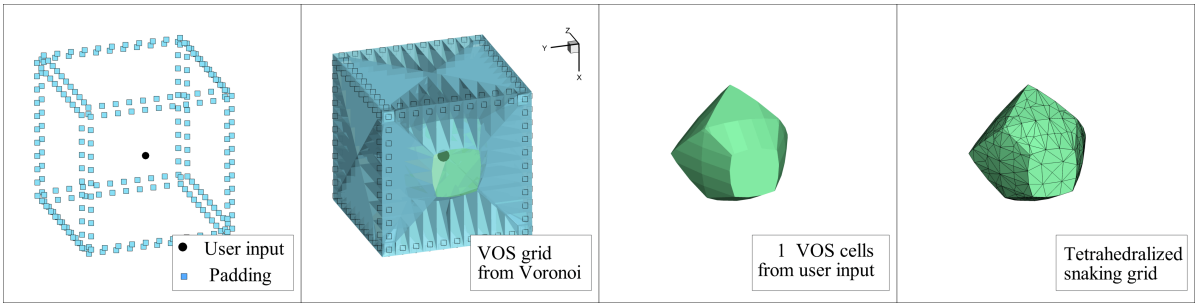
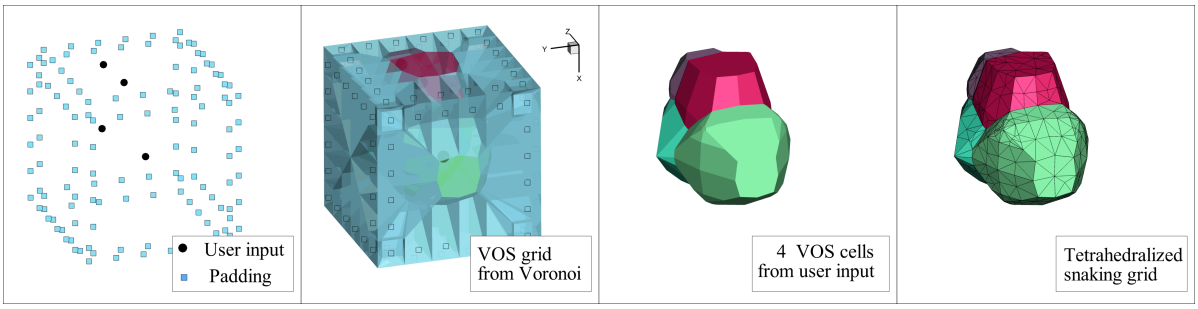
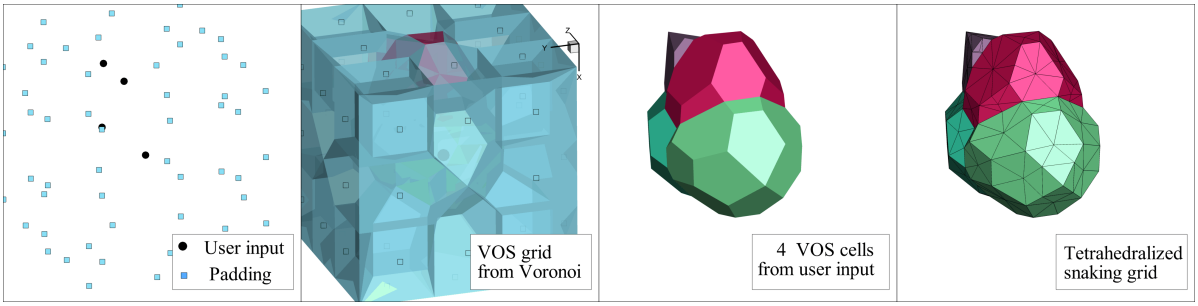


Figure 7.22: VOS mesh generation from a single input point and a padding distance of 0.05.



(a) 4 inputs and padding distance of 0.1.



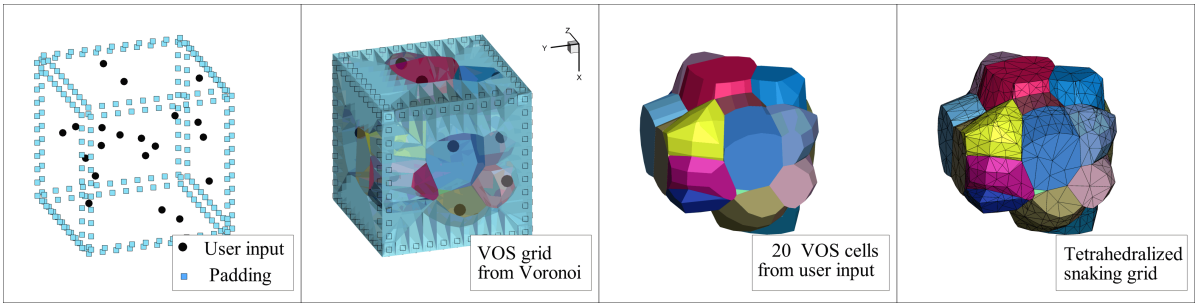
(b) 4 inputs and padding distance of 0.3.

Figure 7.23: Generation of 4 Voronoi cells for various padding distances, the padding distances ensures the cells are closed and the edge of the VOS mesh is convex.

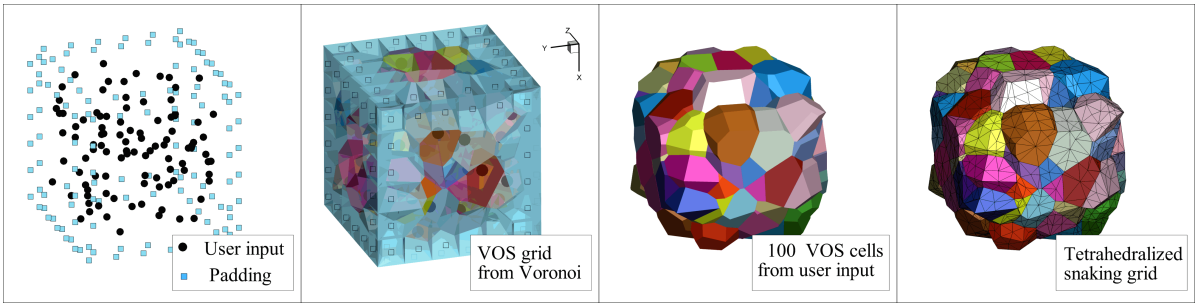
Of those four only the global grid deformations and the Voronoi design cells have been implemented. Examples of the global deformations can be seen in Section 7.2, notably in Figures 7.7 and 7.8 where the x-wise dimension is stretched to achieve an elongated body; and in Figure 7.9 where the span is stretched to achieve a wing-like aspect ratio.

The generation of Voronoi VOS meshes is done in 5 steps which are:

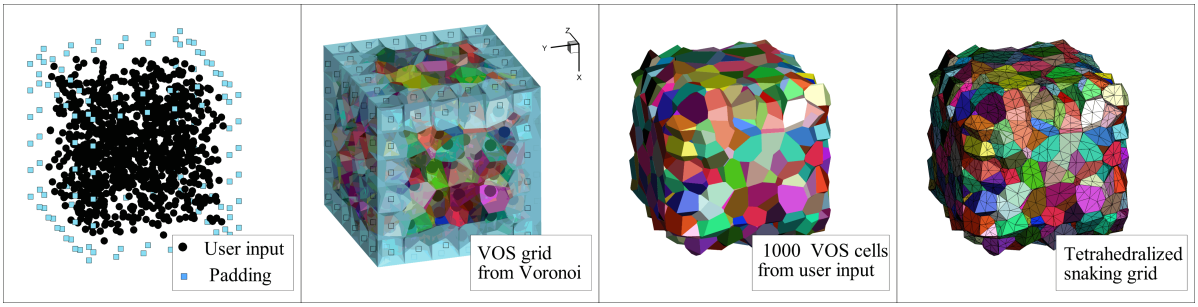
1. **load user input points** the mesh will have a 1 to 1 mapping of point to VOS cells, with the cell occupying the region surrounding its corresponding point;
2. **add padding points** these make sure the VOS mesh can be closed and convex at



(a) 20 inputs and padding distance of 0.05.



(b) 100 inputs and padding distance of 0.1.



(c) 1000 inputs and padding distance of 0.1.

Figure 7.24: Generation of Voronoi design spaces with 20, 100 and 1000 design variables.

the edge of the domain;

3. **generate Voronoi diagram using TetGen [278]** which is used as the VOS mesh;
4. **generate a tetrahedralisation** as the snaking grid on which the r-surface will evolve.

The steps are represented in Figure 7.22 for a single input point (and therefore a single active VOS cell), steps 1 and 2 are combined inside the first image on the left, steps then go from left to right. The impact of changing the padding distance is shown in Figure 7.23 for 4 input points. Examples of Voronoi grids generated with more input points are shown in Figure 7.24.

7.5 Integration into Optimisation Frameworks

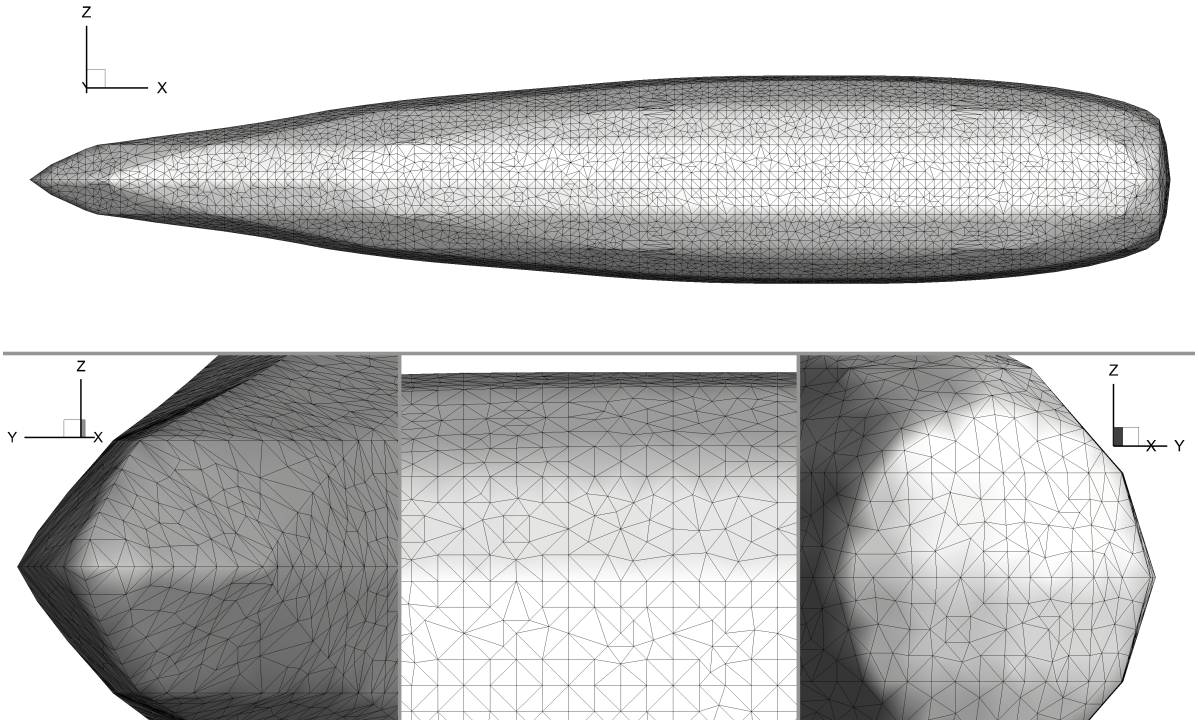
In order to use the 3D-RSVS for aerodynamic optimisation it must be integrated into a framework which supports its topological flexibility. The main bottle neck with current tools are is surface and volume meshing for arbitrary topology; automatic tools are not widely available and traditional finite volume flow solvers are sensitive to mesh quality. Nevertheless, the 3D-RSVS is integrated with SU2 [255] an unstructured flow solver which has been used extensively to tackle aerodynamic shape optimisation (ASO) problems. Recognising that parameterisation methods for three dimensional aerodynamics are usually deformative, a method for the exploiting existing high quality geometries and volume meshes while still exploiting the topological flexibility of the 3D-RSVS is proposed in Section 7.5.3.

7.5.1 Flow Solving for Optimisation of 3D Topology

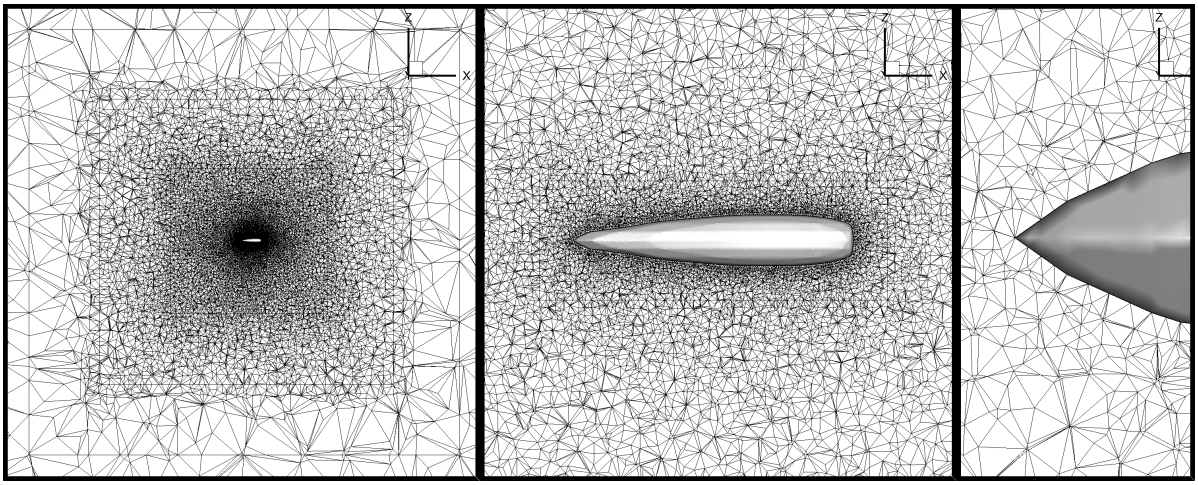
In order to perform optimisation using the 3D-RSVS parameterisation a framework must be capable of running a flow analysis from the geometry without any user input. Meshing of the geometry is done using TetGen [278], the three dimensional version of the Delaunay triangulation tool used in Chapter 6. TetGen generates tetrahedral meshes which can be tailored to a range of applications. In this work TetGen is integrated with the 3D-RSVS to generate volume meshes from the triangulated geometries. Mesh density is controlled in a cut-cell-like fashion: element volume constraints are applied on the mesh through TetGen, with the constraints relaxing away from the body. Figure 7.25 shows the surface and volume meshes for the truncated ogive which was displayed in Figure 7.8.

This approach to mesh generation allows the tuning of element density at the surface to achieve the required resolution of flow properties. The mesh generated using TetGen is suitable for the solution of inviscid compressible flow conditions. An example flow solution using SU2 is shown in Figure 7.26.

Currently these meshes are not suitable to compressible and viscous flows: the Reynolds-Averaged Navier-Stokes (RANS) equations are very sensitive to mesh quality at the surface, and require a smooth prismatic discretisation of the volume in the boundary layer. While tools exist to grow boundary layer meshes around complex geometries, they are often designed for industrial scale applications and require a human in the loop; meshing for RANS is still one of the bottle necks to aerodynamic design [1] and beyond the scope of this thesis. The scope of the challenges is the reason for adopting a modular

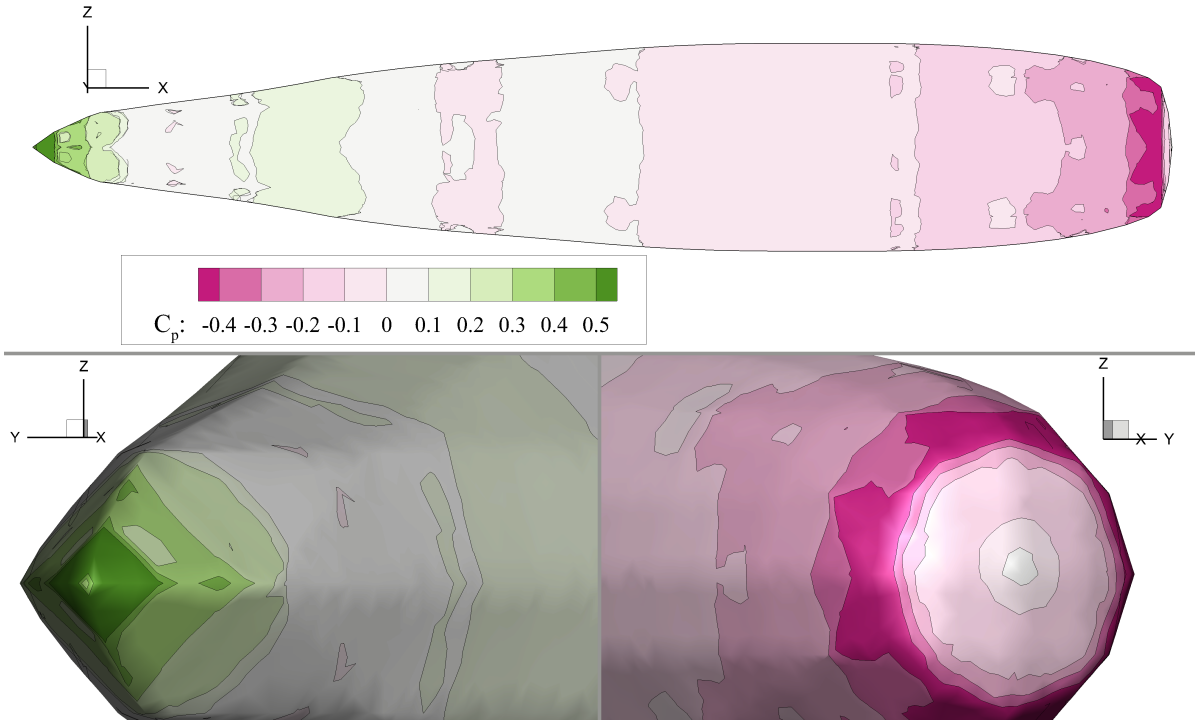


(a) Surface mesh generated by TetGen from a 3D-RSVS geometry.

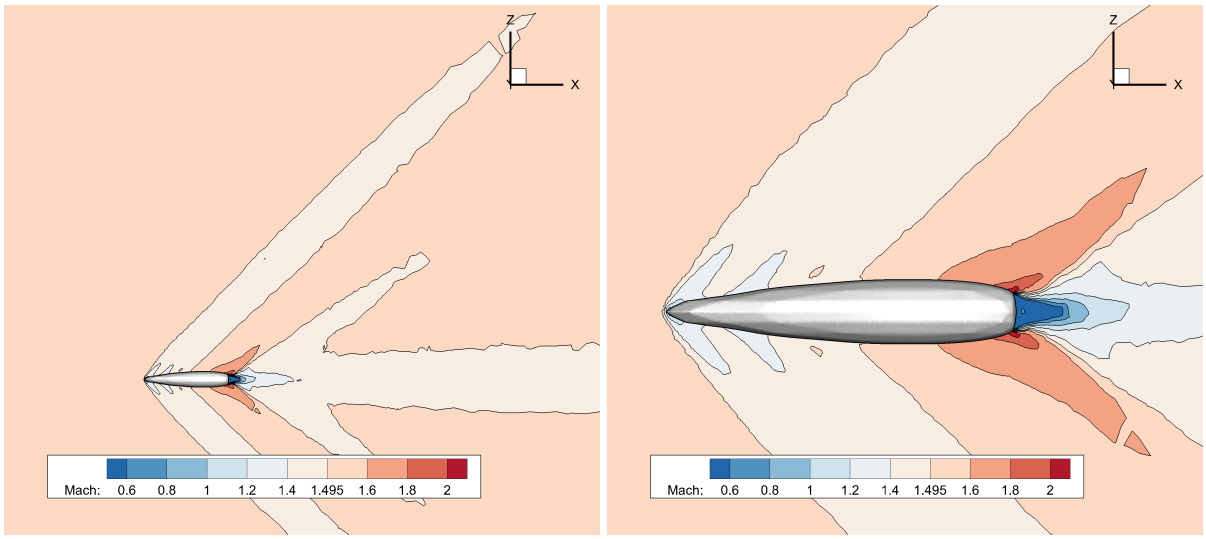


(b) Volume mesh generated by TetGen.

Figure 7.25: Surface and volume meshes for flow analysis of a 3D-RSVS geometry generated by TetGen.



(a) Surface C_p plots from SU2.



(b) Volume Mach number plots from SU2.

Figure 7.26: Surface pressure coefficients and volume Mach number plots on a 3D-RSVS geometry for a free stream Mach number of 1.5 in Euler flow using SU2 ($C_D = 0.00885$)

approach in the development of framework components.

An alternative to the combination of TetGen and SU2 developed here exists commercially in the form of Cart3D [279]. Originally developed by Aftosmis et al. [153], Cart3D implements an efficient three dimensional cut-cell meshing with adjoint based grid adaption to achieve high quality flow results with limited user input. Originally developed for inviscid flows, an interactive boundary layer method has been integrated to provide evaluation of viscous trends in the Cart3D framework [280].

While this system would be suitable for meshing and flow solving, adjoint design sensitivities for the geometry were not commercially available (although available to US federal agencies), ultimately motivating the assembly of TetGen and SU2 to solve compressible flows around complex geometries. Indeed, in three dimensions, the number of design variables and the cost of evaluating a flow solution are such that gradient based optimisation with sensitivities calculated through adjoint flow solvers are the norm. These avoid the excessive computation times that would be the result of agent based optimisation or finite difference gradients.

7.5.2 Integration of the 3D-RSVS with Gradient Based Optimisation

In order for the 3D-RSVS to be effective inside an adjoint gradient based flow solver, the sensitivities of the design variables (the VOS values $\mathbf{V}$) to the aerodynamic forces (C_D, C_L, C_M) need to be computed. To get the sensitivity of the flow to the design variables ($\nabla_{\mathbf{v}}\mathscr{F}$), the chain rule can be used to separate the contributions of the flow solver and the parameterisation process (Equation 7.9). For a flow solution $\mathscr{F}$, calculated on a mesh with vertices at positions $\mathbf{p}$; an adjoint flow solver takes care of calculating the derivative of the aerodynamic coefficients to the position of surface vertices of the aerodynamic mesh ($\nabla_{\mathbf{p}}\mathscr{F}$).

$$\mathscr{F}(\mathbf{p}(\mathbf{V})) = \begin{bmatrix} C_L \\ C_D \\ \vdots \end{bmatrix} \qquad \nabla_{\mathbf{v}}\mathscr{F} = \nabla_{\mathbf{p}}\mathscr{F} \times \nabla_{\mathbf{v}}\mathbf{p} \tag{7.9}$$

The remaining contribution $\nabla_{\mathbf{v}}\mathbf{p}$, is the sensitivity of the surface points to changes of volume fractions used to design the 3D-RSVS geometry. Because the parametrised contour is the result of an SQP process where the volume fraction is a constraint on the design (see Equation 7.2), the calculation of derivatives benefits from a wealth of

previous research into sensitivity analyses for SQP algorithms [89, 90]. Equations 7.10 and 7.11 show how results for the sensitivity analysis of non-linear programs from Buskens and Mauer [90] can be applied directly to the 3D-RSVS to calculate the change in the r-surface due to small changes in VOS.

$$\begin{pmatrix} \nabla_{\mathbf{v}}\mathbf{d} \\ \nabla_{\mathbf{v}}\boldsymbol{\lambda} \end{pmatrix} = -\begin{pmatrix} \mathbf{H}_{\mathbf{d}}\mathscr{L} & \nabla_{\mathbf{d}}\mathbf{h}^T \\ \nabla_{\mathbf{d}}\mathbf{h} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \left(\nabla_{\mathbf{d}}^T\nabla_{\mathbf{v}}\right)\mathscr{L} \\ \nabla_{\mathbf{v}}\mathbf{h} \end{pmatrix} \tag{7.10}$$

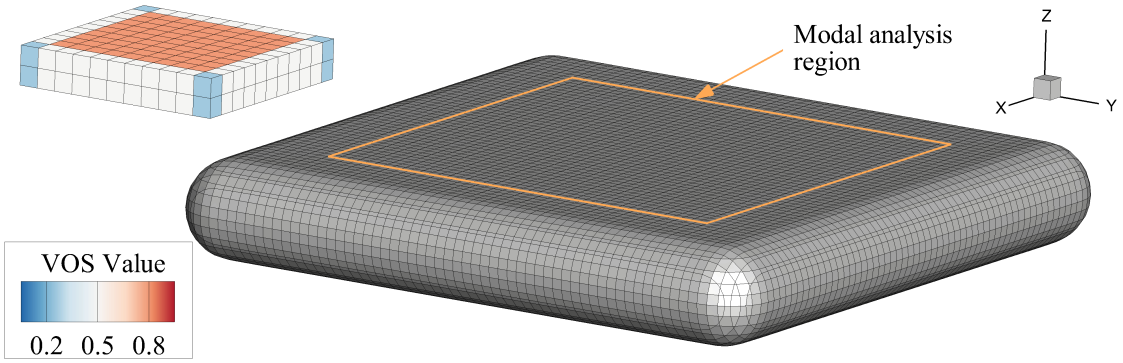
$$\nabla_{\mathbf{v}}\mathbf{p}_i = \nabla_{\mathbf{v}}\left(d_i\Delta\mathbf{g}_i + \mathbf{g}_{i,1}\right) = \Delta\mathbf{g}_i\,\nabla_{\mathbf{v}}d_i \tag{7.11}$$

As was the case for the two dimensional RSVS simply extracting the responses, is likely to be insufficient to ensure the smooth integration with gradient based optimisation. The empirical study in Figure 7.27 shows that the oscillatory response of the RSVS in 2D can also be found in 3D (c.f. Section 4.2). These oscillations are likely to be detrimental to the optimisation behaviours, however these can be resolved the same way they were in 2D: through design variable combination. Figure 7.27d shows that, by grouping cells and changing multiple VOS values as a single "mode", a smoother response is achieved. Defining the exact grouping of VOS cells and their relative response can be pre-calculated or optimal groups can be found for each VOS response layout; these processes were derived in 2D in Section 4.2.

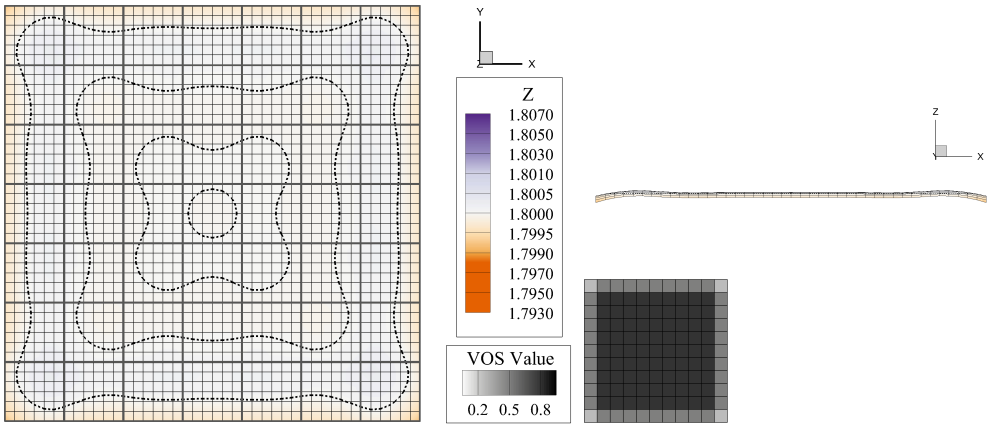
7.5.3 Exploiting Topological Flexibility for Part Design

One drawback of the RSVS is that it is a constructive approach; while this is a necessary feature to enable topological optimisation it means that existing surface and volume meshes cannot be reused. The implication is that the 3D-RSVS cannot be used on large scale industrial geometries, because, while it is theoretically capable to reproduce the entire geometry, in practice it would be an extremely difficult endeavour. Instead, it is suggested that only a portion of a complex geometry would be parameterised, allowing the optimisation of a part without affecting the rest of the geometry; the 3D-RSVS and the original discretisation of the geometry being merged through mesh surgery. Figure 7.28 shows what this approach would look like for the optimisation of winglets for the NASA common research model (CRM) [281].

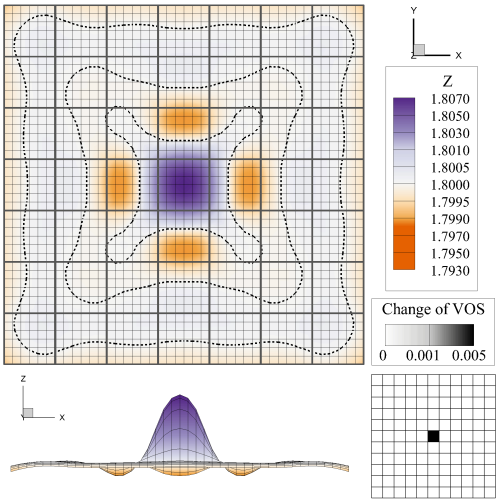
Merging the original surface and the partial surface defined by the RSVS relies on compatible discretisations. Thanks to the flexibility in the shape of VOS cells and topology of the snaking grid; the 3D-RSVS is capable of matching the discretisation of



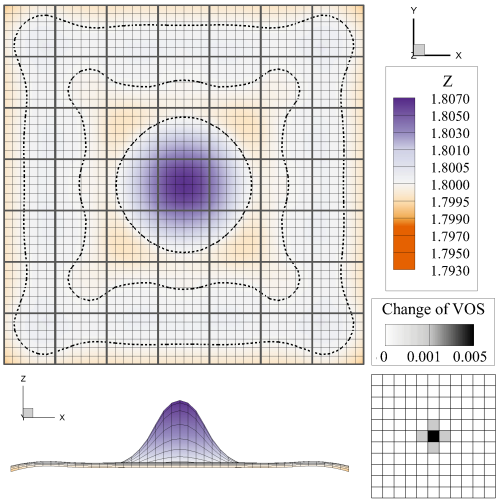
(a) 3D-RSVS surface used for modal analysis. The volume fractions are 0.8 in the flat faces, 0.503 at the edges and 0.26 in the corners.



(b) Analysis of the flatness of the original top surface of the geometry



(c) Oscillatory response of the surface.



(d) Response smoothed with a 20% disturbance in neighbouring cells.

Figure 7.27: Empirical analysis and smoothing of the response of the 3D-RSVS to small changes in volume fractions

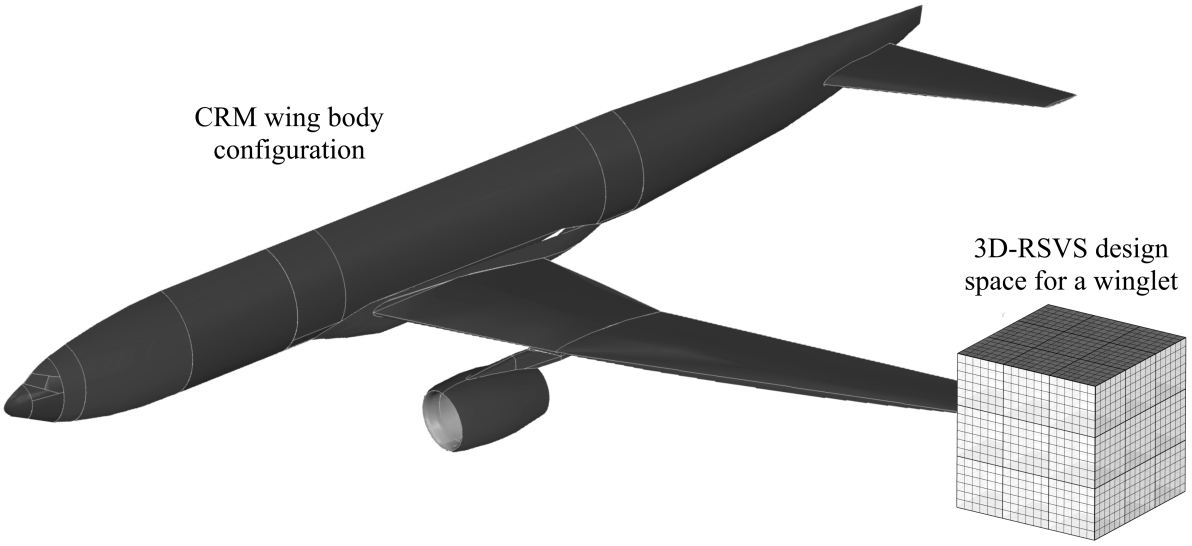


Figure 7.28: RSVS design space embedded in the CRM wing-body-tail configuration. In this configuration only the winglet portion of the design would be modified by the 3D-RSVS.

the original mesh permitting the merge of the geometries at their boundary. Starting from a high quality volume mesh, the region of the design space parameterised by the RSVS can be excavated, the TetGen mesher can then be used to generate a "good enough" mesh which conforms to the RSVS boundary and the cavity inside the original mesh. While this approach would not be suitable for drag prediction purposes, it would provide an indication of relative performance sufficient for optimisation.

7.6 Path to Robust Design Using the 3D-RSVS

While the flexibility afforded by the r-surface and the CMC-like smoothness properties of the 3D-RSVS make it a natural fit for optimisation frameworks some limitations remain with the current implementation. Addressed in the following sections are three key challenges, and possible mitigations, that prevent efficient topology optimisation using the 3D-RSVS. Those are:

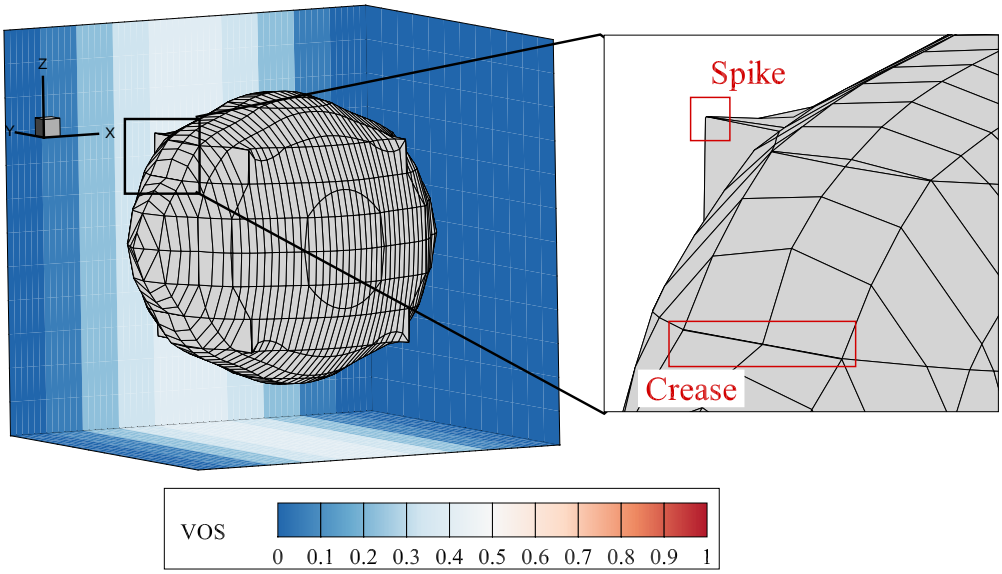
- convergence and stability issues in the form of oscillations due to the SQP step;
- the computational cost of repeatedly solving the quadratic program (QP);
- smoothness of the final geometry being insufficient and only controlled by the snaking grid.

7.6.1 Convergence and Stability

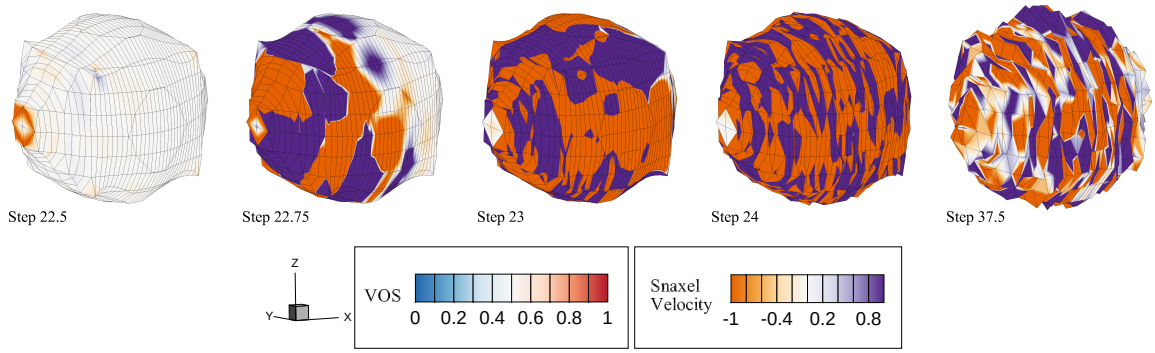
Optimisation using the 3D-RSVS in general and gradient based optimisation in particular, require a smooth response of the parameterisation to small changes of design variables. While the link to CMCs and minimal surfaces suggest that the analytical response of the 3D-RSVS is smooth away from topological changes; it is a significant challenge to get the discrete formulation to reproduce those properties. In its current state the parameterisation does not converge well: without reliable convergence small changes in VOS may lead to significant difference to the profile. Two main types of convergence failures have been observed: premature VOS convergence on profiles with non-minimal area; and profile instability. These are shown in Figure 7.29. These convergence difficulties are the result of the properties of the discrete mathematical program and its derivatives due to the evolution on the snaking grid: the interaction of the floating point mathematics of the SQP and the discrete, integer mathematics of the r-surface are a source of numerical issues.

To understand those issues, it is important to note the difference between the "global" problem solution that is desired; and the "local" solution which exists for every possible discretisation. Indeed, the design variables of the global problem are internal variables of the snaxels (d_i the normalised distance) and these change with each change of topology or each crossing of grid vertices: from a pure optimisation stand-point each of these discretisations is a separate optimisation problem with different design variables. With this realisation, it naturally follows that each of these separate problems, it they can solve the volume constraints, has at least one local minimum defined by those design variables. While many of those are technically "local" minima, in practice they lie at snaking grid vertices ($d_i = 0$ or 1), where the r-surface changes the connectivity for which this geometry is no longer optimal.

In itself this process is not problematic, however it becomes an issue when taking into consideration the properties of the derivatives as neighbouring snaxels converge on the same point. Both in two and three dimensions the objective function ($f(X,Y)$) relies on a square root term whose first derivative is undefined at the origin and whose second derivative is hyperbolic (Equation 7.12). Properties of the edge-length derivatives were discussed in Section 3.5.4 (c.f. Figures 3.30 and 3.31); and motivated the use of a small constant ε to stabilise those derivatives around the origin; this process is shown in Equation 7.13.



(a) Premature convergence.



(b) Unstable oscillations.

Figure 7.29: Examples of the convergence issues of the 3D-RSVS.

$$f(X,Y)=\sqrt{X^2+Y^2} \qquad \frac{\partial f}{\partial X}=\frac{X}{\sqrt{X^2+Y^2}} \qquad \frac{\partial^2 f}{\partial X^2}=\frac{Y^2}{\left(X^2+Y^2\right)^{3/2}} \tag{7.12}$$

$$f_\varepsilon(X,Y)=\sqrt{X^2+Y^2+\varepsilon} \qquad \frac{\partial f_\varepsilon}{\partial X}=\frac{X}{\sqrt{X^2+Y^2+\varepsilon}} \qquad \frac{\partial^2 f_\varepsilon}{\partial X^2}=\frac{Y^2+\varepsilon}{\left(X^2+Y^2+\varepsilon\right)^{3/2}} \tag{7.13}$$

The presence of undefined and infinite values at the boundaries of validity of a discretisation are a significant computational hazard and must be avoided. Now considering the properties of the stabilised derivatives when neighbouring snaxels are lying at the same snaking grid vertex ($X=Y=0$), the first derivative is 0 and the second derivative is a large positive number. Depending on the state of the VOS constraints this point can be an artificial minimum which satisfies the KKT conditions; unfortunately these artificial

Table 7.3: Limits and values of the objective function and the two stabilisation schemes.

	$f(X,Y)$ (eq.7.12)	$f_\varepsilon(X,Y)$ (eq.7.13)	$f_\sigma(X,Y)(eq.7.14)$
h	0	$\sqrt{\varepsilon}$	$\sqrt{\sigma_X^2+\sigma_Y^2}$
$\frac{\partial h}{\partial X}$	$\to \pm 1$	0	± 1
$\frac{\partial^2 h}{\partial X^2}$	$\to \pm\infty$	$1/\sqrt{\varepsilon}$	$\sigma_Y^2/\sqrt{\sigma_X^2+\sigma_Y^2}$

minima are indistinguishable from the actual global minima which may lie close or at a snaking grid vertex. The presence of these minima explains the premature convergence on geometries with sub-optimal area.

Both the original and stabilised second derivatives get very large as both X and Y reduce to 0; this can be the initial cause of profile instability. Indeed these large values sometimes cause very large (10^7) snaxel velocities which if left unchecked would lead to rapid traversal of the snaking grid based on false information. In the current implementation this is avoided by applying aggressive reduction of the step size which inflicts an additional convergence penalty on the 3D-RSVS with no guarantee that it will not lead to some oscillations.

$$\begin{gathered}\text{Defining } X_\sigma = X + sign(X)\sigma \text{ for } X \text{ and similarly } Y_\sigma \text{ for } Y \\ \text{where}: sign(x) = 1 \ \forall x \in \mathbb{R}^+ \text{and } sign(x) = -1 \ \forall x \in \mathbb{R}^- \\ f_\sigma(X,Y) = \sqrt{X_\sigma^2+Y_\sigma^2} \qquad \frac{\partial f_\sigma}{\partial X} = \frac{X+sign(X)\sigma}{\sqrt{X_\sigma^2+Y_\sigma^2}} \qquad \frac{\partial^2 f_\sigma}{\partial X^2} = \frac{Y^2+2\sigma|Y|+\sigma^2}{\left(X_\sigma^2+Y_\sigma^2\right)^{3/2}}\end{gathered} \tag{7.14}$$

A theoretically better stabilising equation is suggested in Equation 7.14: by avoiding a rapid change of the first derivative from approximately 1 to 0 as X tends to 0 the second derivative can be better behaved. Limits and values of these approaches to objective function stabilisation are compiled in Table 7.3 and their behaviour is shown in Figure 7.30. By having a non 0 gradient at the origin the frequency of spurious minima may be decreased. The challenge of this formulation remains the definition of the $sign(0)$ which is both -1 and 1 depending on the approach direction. While not currently implemented because of this perceived limitation, this formulation of the objective could improve the reliability of 3D-RSVS convergence.

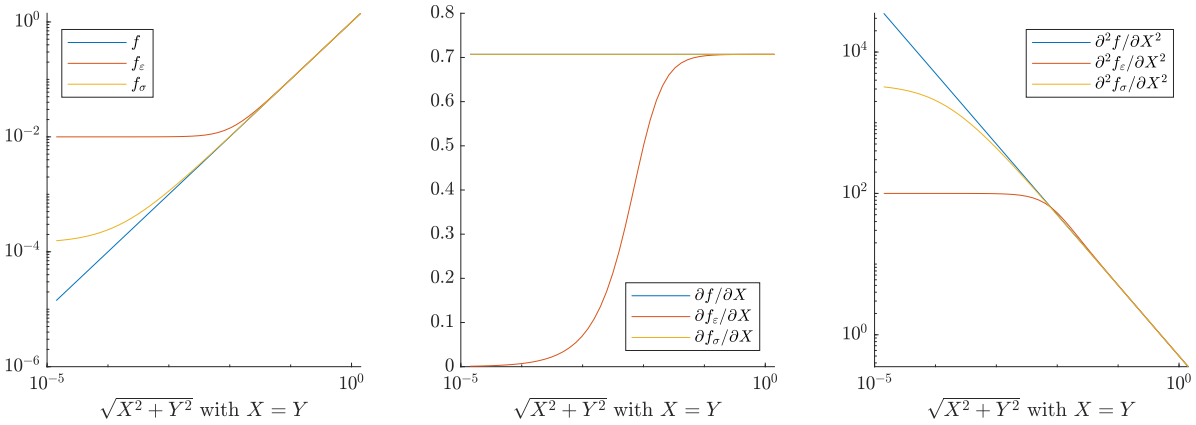


Figure 7.30: Evolution of the natural objective and the two proposed stabilising methods, and their derivatives close to $X = Y = 0$

7.6.2 Solution of the Quadratic Program (QP)

SQP methods are well known for their quadratic convergence on a constrained minima, even when using approximations of the Hessian; this property is very powerful and makes them the obvious choice to tackle optimisation of smooth functions. The "Surface Evolver", certainly the most widely used and studied discrete CMC solver, uses a Hessian based quadratic solver, stating that local convergence may be achieved in as low as four steps [275]. In the case of the 3D-RSVS, for large numbers of design variables (10^3 snaxels and above) the computational cost of the parameterisation is dominated by finding sequential solutions to the QP. Normally Newton based methods make up for this limitation by requiring very few steps to converge; unfortunately because of the convergence challenges detailed in the previous paragraphs, the number of steps for RSVS convergence can be large (hundreds).

In order to mitigate the computational cost of the QP solution the structure of the Hessian matrix can be exploited to accelerate the solution process. Indeed the 3D-RSVS system is sparse, with the width of the central diagonal unchanging with system size. This property means that sparse matrix algebra can be used to speed up and reduce the memory footprint of the QP solution process; current experiments suggest that the cut-off for a reduction in computational cost is between three and four thousand snaxels.

Algorithms used for sparse mathematics are also more easily parallelisable than their dense equivalents; the structure of the problem means that regions of the design space can be more easily separated. The matrix mathematics library used to solve the 3D-RSVS QP, Eigen, supports parallel sparse solvers out of the box [282].

While the cost of the QP can be reduced, for large surfaces it is likely to remain

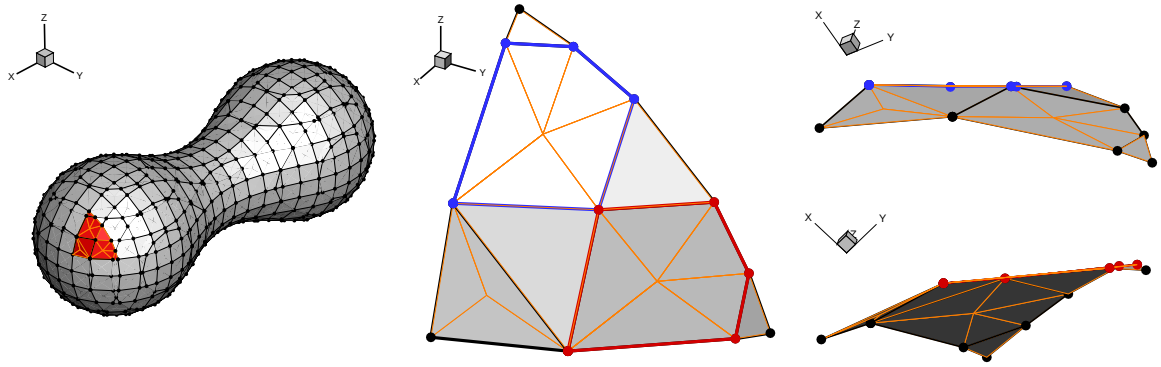


Figure 7.31: Examples of flat faces in curved region of the design space.

a bottle neck; as such it may be necessary to consider other optimisation algorithms. A host of methods have been used and specifically tuned for large scale constrained minimisation: sequential linear programming, constrained conjugate gradient methods, interior point methods, multipoint exponential approximation (MPEA) or the quadratic multipoint exponential approximation (QMEA) [283].

The main limitation of non-quadratic methods is that they do not specify a step length, which can lead to slow convergence if a robust trust region method is not available. It is interesting to note that the limits on step size already in place in the snaking algorithm implicitly set a trust region far from the optimum. Since the benefits of quadratic programming lie mostly close to the optima, it may be beneficial to swap dynamically between algorithms; exploiting the simplicity and cheap cost of linear solvers in the far field, only to use the SQP to arrive rapidly at the minimum once the algorithm is in the neighbourhood of a minimum.

7.6.3 Smoothness of the Output Surface

Since the r-surface evolves on a predetermined grid according to specific rules, the surface vertex distribution is not explicitly controlled. In practice, this leads to surface triangles with poor aspect ratios and wildly different sizes which is generally not conducive to robust physical modelling. While the surfaces generated by the 3D-RSVS are in general smooth, the lack of explicit control over the centroid of cells often leads the parameterisation to generate flat faces even in curved regions of the geometry. This behaviour creates artificially flat groups of triangles in a part of the geometry which, according to the governing equation, should have a smooth curvature. Figure 7.31 shows an example of large, flat, pentagonal faces in a region of otherwise uniform curvature.

The flattening observed in Figure 7.31 could be resolved by providing the centroid with its own design variable, limiting movement to be in a normal direction to the face that spawned it. This additional design variable, would enable the SQP to smooth out any panelling created by the interaction of the SQP and the centroid calculation process. The increase in number of design variables due to the addition of almost 1 per face, would be offset by a reduction in the density of the snaking grid.

While it reduces the impact of the triangulation, this approach does not improve the control over element size and shape. This approach also has a major downside in that it looses guarantees of water-tightness and the absence of self intersection.

7.6.4 De-restricting the Surface

The benefits of using a restricted surface are undeniable when it comes to resolving the topology of a geometry and the connectivity between VOS cells. The efficiency and reliability of the integer collision detection process make it possible to consider very complex geometries with little computational cost; however that is not the only requirement of a three dimensional topology parameterisation method. The accumulation of issues at the interface between the integer mathematics of the restricted surface method and the continuous solutions of the governing equation suggest that a radically different approach may be warranted. Indeed convergence, stability and smoothness difficulties are all, in part, attributable to the use of a restricted surface. This is not to say that it is not possible to solve these issues within the restricted framework, but it may be more efficient and robust to consider alternatives.

Inspired by the “Surface Evolver” [274] and its descendants, the possibility to take the r-surface off the grid is considered. De-restricting the surface would lose the very efficient topology control, but gain smoother profile progression. For this reason a two step process is proposed: first the 3D-RSVS is evolved as described in this chapter to resolve topology and cell connectivity; then it is separated from the snaking grid to finish its convergence without topology change. By separating the surface from the grid, element distribution can be controlled; convergence is no longer hampered by changes of design space due to changes of connectivity; and fewer QP solutions will be necessary.

Using the idea of de-restricting the surface, the complete 3D-RSVS process would become:

1. **Resolve topology through the restricted surface:** Use the existing process on a coarse snaking grid to rapidly converge the topology and cell connectivity.

2. **Place snaxels in "surface space":** generate a pseudo snaking grid which is only a set of edges normal to the current surface. This allows: maximum code re-use; the creation of implicit trust-regions, through the length of each edge; and enforcement of the VOS boundary, ensuring that the snaxels stay on VOS cell boundaries. At this stage, the surface may be refined.
3. **Evolve the "de-restricted" surface:** iterate the position of the snaxel along the pseudo-grid edges, according to the SQP. The edges of the pseudo-grid need not be constant through the convergence process allowing simultaneous resolution of the constraints, minimisation of the objective and redistribution of surface elements. A method similar to the CMC-CVT of Pan et al. [272] is suggested.

7.7 Summary

In its current state the 3D-RSVS cannot compete with existing parameterisations, it does not have the robustness expected of such methods. Fortunately the causes of these limitations are understood, evidence shows that they are not intrinsic to the formulation, and, ways to resolve them have been outlined.

The original goal of three dimensional topology optimisation for aerodynamic bodies is very close: the necessary flow solvers, meshers and optimisation methods are all already in regular use even. While the missing link remains a parameterisation which will describe arbitrary topologies with sufficient smoothness, compactness and flexibility for aerodynamic application; this chapter has shown the feasibility of parameterising shape and topology to an aerodynamic optimisation standard.

Beyond its native application in aerodynamic parameterisation, the scope of the 3D-RSVS is much broader. It appears that the formulation of the 3D-RSVS lets it bridge the gap between level set methods and traditional density methods; while establishing a footing for the parameterisation in differential geometry. The natural formulation of the RSVS gives rise to very intuitive behaviours, and the simplicity of the definition leads to a very general method. The intuitiveness of the method makes it suited for applications in which it is controlled by a designer, notably for prototyping. In fact the 3D-RSVS has the potential to be a framework for the design of smooth surfaces: new types of constraints, and, changes of objective function could impart on the 3D-RSVS most of the features of a CAD framework.

CHAPTER 8

Conclusions and Recommendations for Future Work

This thesis has developed the methods required for topology optimisation of external aerodynamics. Behind this titular endeavour lies the desire to make physical simulation tools a routine part of a modern and flexible design process. Non-linear physical simulation tools are challenging to a designer as their behaviours can be unintuitive and are not intrinsically parametric: it depends on a geometry which does not obey a unique or natural parameter space. In order to navigate the domain of possible solutions, engineers rely on 'expert knowledge': an incomplete understanding of those non-linear behaviours.

The recent democratisation of analysis tools and improvements in their flexibility means that computational fluid dynamics (CFD) is no longer the sole remit of large aircraft manufacturers. The use of aerodynamic optimisation beyond the traditional CFD user-base is restricted by the limited geometric flexibility afforded by aerodynamic shape optimisation (ASO) methods. The restricted snakes volume of solid (RSVS) and its integration with the multi-level subdivision optimisation (MLSO) are tools which have the robustness and flexibility to tackle this growing range of physics-driven design problems. While most cases tackled were in the topology optimisation of external aerodynamics, it is hoped that the developments of this thesis can have broader applicability.

8.1 Exploration of the Aerodynamic Design Space

Included in a modular ASO framework, the versatility of the RSVS enabled the optimisation of supersonic geometries under area constraints in inviscid flows. The RSVS was able to exploit flow physics to outperform known analytical optima and previous shape optimisation results. In particular the use of hierarchical design variables permitted effective optimisation through a range of geometric behaviours due to changes in constraint values. These results extend to topology optimisation: the RSVS enabled the design of optimised multi-plane profiles for a range of area constraints, beyond what Busemann bi-planes were known to achieve. Further improvements in the thoroughness and efficiency of the optimisation was enabled by the development of the combined RSVS-MLSO framework. These improvements in efficiency permit a much better understanding of the aerodynamic design space to be gained. The integrated RSVS-MLSO approach revealed multi-modality, discontinuous flow behaviour, and, optima with periodic geometric patterns.

The study of academic optimisation cases highlight that it is a feature of aerodynamic optimisation that complex behaviour should be expected. There are few reasons to believe that equally intricate behaviours would not arise in the more complex cases tackled by designers for industrial applications. While ideally new algorithms would be developed on problems with real world applications, the tractability of academic problems is necessary to the development and validation of new methods.

8.1.1 Brittle Optimisation Results

The pathological behaviour of the flow for optimal multi-body supersonic profiles is similar to that exhibited in the NACA0012 ADODG benchmark Case 1: an extreme sensitivity of shock patterns to geometric change makes for very brittle optimisation solutions. These observations confirm the notion that single-point inviscid drag minimisations tend to degenerate designs with very narrow windows of performance: such results cannot be used ‘as is’ in the design process and must be enhanced with more information.

Clear paths exist in the literature to improve the problem of these ill-conditioned results. From a physical modelling approach, minima under Reynolds-Averaged Navier-Stokes (RANS) flow have generally smoother responses thanks to the dissipative effect of viscous terms. However a large challenge remains in obtaining flexible, unsupervised and automatic meshing for RANS flow solvers. Optimisation methods can also reduce the sensitivity of optima: ‘robust optimisation’ explicitly, and multi-point optimisation in

practice; require a wider window of efficient operation from the optimised design.

8.1.2 Multi-Modality in Drag Minimisation

The aerodynamic ill-conditioning of these cases also impacts the modality of the overlaid optimisation cases: the sensitive transition between flow states leads to a high degree of apparent multi-modality. Because the discontinuity in the aerodynamics breaks the assumptions that underpin gradient based optimisation theory, optimality of the final solutions cannot be formally evaluated. In order to work around this discontinuity two methods were suggested in Section 6.5: development of explicit constraints, or, the use of surrogate models. Both these methods were shown to be effective on an analytical problem with similar features, and could help further improve understanding of the aerodynamic design space around multi-body profiles and flow topology transitions.

While the number of true minima existing for a given topology is uncertain, it is clear that there exists at least one local optimum for single body geometries and one for multi-body geometries. In design optimisation, the object does not neatly exist in a single parameter space: its representation and evolution is the result of a parameterisation process. As such the number of local minima of a given problem is indissociable from a specific parameterisation: a good parameterisation does not create spurious local minima, and, instead, discriminates between designs with different physical behaviours.

Using the RSVS-MLSO, different flow patterns were clearly observed in single and multi-body optimisation: it appears that each topology has at least one local minima. While optimal profiles exhibit different topologies and geometries, the flow patterns that result are very similar; raising the question whether these truly represent separate solutions to the underlying physical problem. The topological response of the drag minimisation of supersonic geometries under area constraints can be seen as cyclical, with the same high performing solution appearing at each topology. To further explore these cases quality diversity and niching approaches are recommended: they would enable the classification of these optima in a much more systematic fashion.

8.2 Generic and Compact Shape Control

The challenging nature of compressible aerodynamic topology optimisation stems from the dichotomy between the material distribution nature of topology definition and the boundary driven behaviour exhibited by aerodynamic optimisation. Aerodynamic topol-

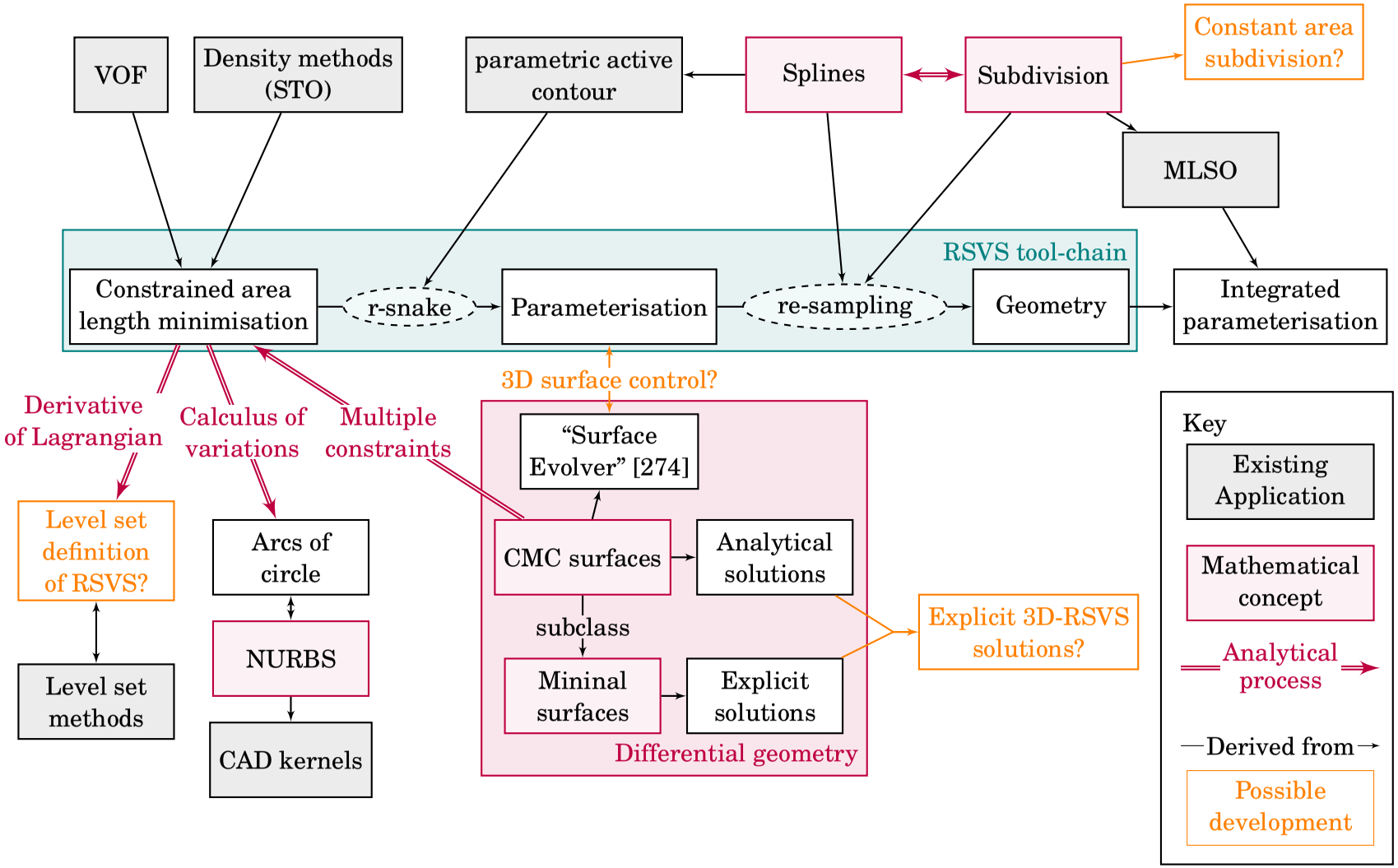


Figure 8.1: Summary of the relationships between the RSVS and other geometric tools and optimisation methods.

ogy optimisation is a relatively new field of research which has been most successful where the aerodynamic problem can be naturally phrased in the same form as structural topology optimisation (STO) cases: as a problem of material distribution. This formulation is possible for most incompressible and small scale flows, where viscosity effects dominates; however it does not extend to external flows where compressibility and shock waves are present. In the compressible regime optimisation of geometries is a boundary design problem, where small changes of the surface can have a very large impact on the observed forces.

To enable topology optimisation for external aerodynamics, the RSVS was developed by drawing from the insights of boundary design parameterisations used in ASO, as well as existing topologically flexible parameterisation methods. The RSVS, thanks to its formulation, has desirable intrinsic properties as a boundary design tool: G_1 continuity; an explicit equivalence to NURBS; and a smooth local response. Uniquely, the RSVS combines these properties with a definition naturally suited to material distribution problems. Figure 8.1 presents a summary of the links the RSVS has to other geometry control methods. Beyond the specific formulation studied in this thesis, the RSVS is envisaged as a modular framework for the definition of smooth, intuitive and flexible parameterisation methods: modification of the objective function could generate geometries with higher order derivative continuity.

The modular framework developed around the RSVS proved its versatility by tackling a standard ASO benchmark case (ADODG Case 1-NACA0012); geometric matching of aerofoils; and topology optimisation of structural cases. The RSVS is a versatile tool which enables optimisation and design beyond a single physical analysis space. In this scope much of the flexibility of the RSVS in terms of anisotropic volume of solid (VOS) cells remains under-exploited; and could lead to efficient combined aero-structural topology optimisation.

The restricted surface volume of solid (3D-RSVS) was developed to bring topological flexibility to three dimensional aerodynamic design problems. A significant contribution of this thesis was the development of the restricted surface (r-surface) a tool for the topological evolution of water-tight surfaces, in three dimensions, and possibly in more. Like the 2D parameterisation, the 3D-RSVS relies on exact volume fraction matching specified on an arbitrary grid. A strong relationship between the formulation and empirical behaviour of the 3D-RSVS and constant mean curvature (CMC) surfaces was described. This link to CMC surfaces suggests approaches to improve the computation of the 3D-RSVS surfaces: current limitations of the 3D implementation are not intrinsic

and can be overcome using the methods common in existing CMC solvers.

While the three dimensional parameterisation has been developed, work remains on its exploitation for aerodynamic topology optimisation. Integration into an optimisation framework was outlined but not used due to reliability issues in the solution process of the 3D-RSVS itself. Many of the tools that made the 2D parameterisation very effective have not been tested on the 3D-RSVS despite its similar properties; for example, the VOS layout flexibility of the RSVS, through refinement, enabled automatic case specific adaptation in two dimensions; this should be extended to three dimensions.

8.3 Recommended Extensions of the RSVS

In general, in both the two and three dimensional parameterisation the ability of the RSVS to use arbitrary VOS layouts and snaking grids has been underused. Beyond VOS cell splitting during refinement, the grid itself can be deformed to achieve FFD-like control, which would allow smooth transition though the geometric discontinuity caused by topology change. Beyond Cartesian layouts, the capacity to have stable and smooth geometries from complex VOS layouts is one of the most powerful features of the RSVS and 3D-RSVS parameterisations: it allows optimisation under shape constraints by using conformal design variable layouts. Not necessarily needed for aerofoil design, the ease of configuration makes it well suited for less studied, niche, design optimisation problems. The intuitive behaviour of the parameterisation and its ability to naturally design water tight surfaces of arbitrary topology make it the ideal tool for the preliminary design of objects under physical loads and constraints.

In order for new, automated, tools to meaningfully improve the design process, they must allow the thorough exploration of the space of physical behaviours and allow a designer to distil that information into design principles. The RSVS, 3D-RSVS, and, integrated parameterisation methods are all steps in this direction: used as interactive tools, they could enhance CAD kernels; allowing the definition of complex surfaces, and, streamlining numerical exploration and optimisation. In optimisation, much work remains in the application and validation of these methods to more complex cases, notably with component constraints and partial parameterisation of design spaces. Resolving these challenges would streamline the engineering of physical systems; democratising physics driven design in the information age.

APPENDIX A

CONVERGENCE TEST ON THE ROSENBROCK FUNCTION

A.1 Formulation of the "Rosenbrock Banana Function"

The "Rosenbrock banana function" is defined in two dimensions by Equation A.1 and in N dimensions by Equation A.2. In two dimensions it has strictly 1 minimum at (1, 1). For higher dimensions a vector of design variables of all ones will always be a root and global minimum but other local minima may exist.

$$f(x,y) = (a-x)^2 + b\left(y-x^2\right)^2 \tag{A.1}$$

$$f(\mathbf{x}) = \sum_{i=1}^{N-1}\left[100\left(x_{i+1}-x_i^2\right)^2 + (1-x_i)^2\right] \quad \text{where} \quad \mathbf{x} = [x_1,\ldots,x_N] \in \mathbb{R}^N \tag{A.2}$$

A.2 Note on the Optimisers Used

Most of the optimisers tested in Section 2.1.2 were those provided by the python package *scipy*, with some of them implemented as extensions to the package by the author. The optimisers used "as is" from the package are:

- Nelder-Mead;
- Conjugate gradient;
- Newton with line search;
- BFGS;

- L-BFGS SQP.

The others were implemented by the author:

- Newton;
- Steepest descent.

Table A.1: Convergence information of the optimisers on the 2D Rosenbrock function.

Algorithm	**Function Evaluations**	**Iterations**	$log10(f(x^*))$
Nelder-Mead	233	123	-13.044
Conjugate Gradient (PRP)	57	20	-20.233
Steepest Descent	5638	5411	-9.938
BFGS	45	34	-13.611
Newton-LS	113	95	-16.818
Newton	21	20	-16.560
L-BFGS SQP	47	35	-13.505

Table A.2: Convergence information of the optimisers on the 5D Rosenbrock function.

Algorithm	**Function Evaluations**	**Iterations**	$log10(f(x^*))$
Nelder-Mead	1111	702	-12.590
Conjugate Gradient (PRP)	271	150	-11.022
Steepest Descent	4195	3880	-9.793
BFGS	69	58	-13.194
Newton-LS	64	50	-13.084
Newton	39	38	-8.207
L-BFGS SQP	49	43	-10.918

Table A.3: Neighbourhood convergence information of the optimisers on the 10D Rosenbrock function.

Algorithm	**Function Evaluations**	**Iterations**	$log10(f(x^*))$
Nelder-Mead	1104	728	-11.581
Conjugate Gradient (PRP)	247	147	-10.644
Steepest Descent	3514	3287	-9.834
BFGS	29	17	-13.235
Newton-LS	10	9	-9.096
Newton	8	7	-7.719
L-BFGS SQP	25	20	-9.474

APPENDIX B

DEVELOPMENT OF A CONSTANT AREA SUBDIVISION

Subdivision curves are a geometric tool, used in modelling and animation to efficiently generate and control smooth surface. They have been successfully used in aerodynamic shape optimisation (ASO) for parameterisation by Masters et al. [176, 263]. The multi-level subdivision optimisation (MLSO) which was used in conjunction with the restricted snakes volume of solid (RSVS) to form an efficient "integrated framework" relies on the naturally smooth and hierarchical formulation of Subdivision curves for parameterisation. In this thesis Subdivision curves were also used in Section 5.2 to re-sample restricted snakes (r-snakes) to achieve surface discretisation suitable for computational fluid dynamics (CFD).

Despite their desirable properties, subdivision curves have a limit when integrating with the RSVS, refinement of a profile leads to small changes of area and violation of the area constraint. Before the derivation of an explicit analytical form for the RSVS, it was thought that subdivisions, thanks to their flexibility would be the most straightforward way to re-sample arbitrary geometries. Beyond the goal of re-sampling, the existence of a subdivision area matching process could establish a more natural link between density based methods, non-uniform rational B-Splines (NURBS) and subdivision.

In order to maintain the exact area matching property of the RSVS a smooth subdivision scheme which maintains internal area is devised. As this appendix details, this property is possible since, the calculation of area can be phrased as a matrix Equation 3.35.

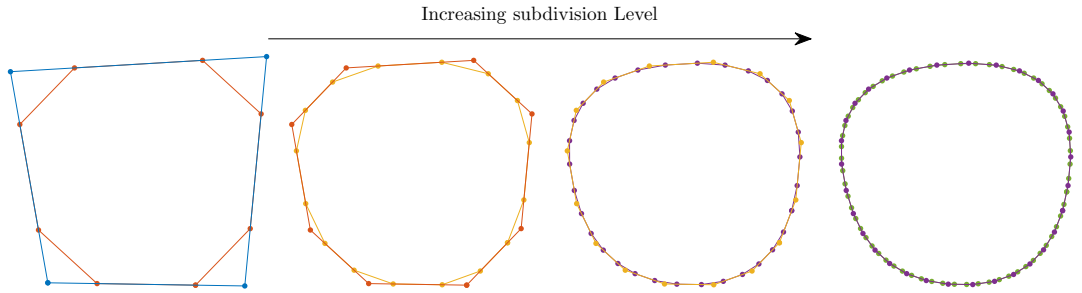


Figure B.1: Four levels of subdivision of a four point control polygon.

B.1 Subdivision for Smooth Surface Point Distributions

A range of subdivision schemes have been developed by the computer graphics community to generate smooth shapes from a small number of control points. Chaikin and B-splines subdivision curves have successfully been used in aerodynamic parametrisation by Masters et al. [176]. Subdivision schemes rely on the sequential linear combination of control points to create a new, larger, set of points with pre-determined properties. The ability of these methods to deal efficiently with an arbitrary set of points makes them ideal for the refinement of profiles generated by the snaking process. The general formulation of subdivision schemes is presented in Equation B.1.

$$\mathbf{p}_{k+1} = \mathbf{S}_k \mathbf{p}_k \tag{B.1}$$

Where $\mathbf{p}_k$ are a set of points transformed by matrix $\mathbf{S}_k$ into a larger set of points $\mathbf{p}_{k+1}$. $\mathbf{S}_k$ is a matrix built from repeating a sub-matrix known as the mask. The mask governs the properties of the vertex averaging process and can vary depending on the goal of the application. The methods named above are primarily designed to generate smooth shapes, but other properties can be designed into the process to suit other applications. To leverage the maximum potential of Volume of Solid parametrisation and subdivision curves the development of a subdivision process conserving internal area is explored in the following Section B.2.

In addition to the progressive smoothing and refinement of profiles a limit curve to the subdivision process can be calculated. For stable subdivision schemes a limit curve can be calculated which corresponds to the continuous curves on which the points lie as k tends to infinity.

$$\mathbf{p}_\infty = \lim_{k\to\infty} \mathbf{p}_k = \cdots \mathbf{S}_{k+1}\mathbf{S}_k\mathbf{p}_k \tag{B.2}$$

$$\mathbf{p_f} = \mathbf{R}_s \Phi_{\infty,N} (\mathbf{S}_N \mathbf{S}_{N-1} \cdots \mathbf{S}_1)\mathbf{p}_1 \tag{B.3}$$

Results from eigenanalysis allow the projection of a given subdivision level onto the limit curve [284]. To select the distribution, these projected points are interpolated using cubic-splines and re-sampled using a suitable point distribution. This process is shown in B.3 above, where $\mathbf{p}_1$ are the original points, $(\mathbf{S}_N\mathbf{S}_{N-1}\cdots\mathbf{S}_1)$ is the standard subdivision process; $\Phi_{\infty,N}$ is the result of the projection to the limit curve at the N^{th} subdivision level; and $\mathbf{R}_s$ is the correlation matrix of the interpolation scheme.

The subdivision re-sampling process outlined above does not enforce the area constraint which is used as the design variables: it tends to generate a profile with a slightly smaller internal area. While this does not affect the quality of the control of the overall algorithm: it remains smooth and reliable; it does affect the ability to directly impose area constraints on the parameterisation. To maintain this desirable property a constant area subdivision process is developed in Section B.2. Previous work using subdivision curves, to parameterise and re-sample aerodynamic geometries have only needed to generate aerofoils in 2-dimensions; this allowed the use of the well known "cosine distribution" to determine the correct point locations. In topology generation there is no guarantee that an object will have a shape compatible with standard point distributions. For a robust re-sampling of arbitrary topologies a generic equivalent is developed in Section 5.2.2.

B.2 Derivation of a Constant Area Subdivision Process

To maintain the exact volume control of the RSVS any profile re-sampling step needs to match the property of the parameterisation. The development of a constant area subdivision process allows the application of area constraints directly on the design variables, making them extremely simple to manage for an optimiser. For a subdivision scheme to maintain constant profile area the matrix form of Green's theorem is required (Equation 3.34, re-stated bellow).

$$A = \frac{1}{2}\mathbf{p}^T\mathbf{R}_A\mathbf{p} \tag{B.4}$$

Let Equation B.4 be considered for an arbitrary set of points $\mathbf{p}_k$ and its refined set $\mathbf{p}_{k+1}$.

$$A_k = \frac{1}{2}\mathbf{p}_k^T \mathbf{R}_{A,k} \mathbf{p}_k \tag{B.5}$$

$$A_{k+1} = \frac{1}{2}\mathbf{p}_{k+1}^T \mathbf{R}_{A,k+1} \mathbf{p}_{k+1} \tag{B.6}$$

For a constant area method A_k and A_{k+1} are equal; taking this into account and recalling the general subdivision equation between two sequential sets of points(Equation B.1), a relationship between area at the k^{th} and $k^{\text{th}+1}$ subdivision step can be derived. These lead to Equation B.8: the necessary condition for any subdivision scheme maintaining area.

$$\begin{aligned} \frac{1}{2}\mathbf{p}_k^T \mathbf{R}_{A,k} \mathbf{p}_k &= \frac{1}{2}(\mathbf{S}_k \mathbf{p}_k)^T \mathbf{R}_{A,k+1} (\mathbf{S}_k \mathbf{p}_k) \\ &= \frac{1}{2}\mathbf{p}_k^T \mathbf{S}_k^T \mathbf{R}_{A,k+1} \mathbf{S}_k \mathbf{p}_k \end{aligned} \tag{B.7}$$

$$\mathbf{R}_{A,k} = \mathbf{S}_k^T \mathbf{R}_{A,k+1} \mathbf{S}_k \tag{B.8}$$

To design the subdivision scheme that will fulfil this condition requires an appropriate layout of non-zero terms in the subdivision matrix. The set of non-zero terms is called the mask (or stencil) of the subdivision process. The width of the mask controls the region of influence of each point and needs to provide a sufficient number of degrees of freedom to allow at least one solution to Equation B.8. Once a mask is selected the matrix operation is performed producing a system of polynomial equations. These equations were solved programmatically allowing quick analysis of a range of mask sizes and layouts. These systems can be under-constrained and lead to families of solutions. This can be exploited to create subdivision schemes with a range of characteristics. Desirable properties other than area conservation include smoothness and convergence.

To define a mask, the first parameter to select is the number of 'children' per node. Initial investigations seem to indicate that no binary subdivision scheme will provide a stable solution to the area condition. The first stable solutions appear for ternary schemes (every point is split into three). Once the number of splits is chosen it is necessary to select the range over which each point will have an influence. Preliminary observations show that each point must at least have an influence two steps away for the volume to be maintained.

In order to find a working mask the matrix equation needs to be built using a suitable layout of coefficients. The layout in Equation B.9 was used to generate the scheme in

Equation B.12 and the results in Figure B.2. These results correspond to a ternary scheme, meaning that each point is split in 3 at each refinement step. This layout was the only one for which stable, smoothing solutions were found while investigating the solutions of the system.

$$\mathbf{m} = \begin{bmatrix} f & e & d & c & b & a & a & b & c & d & e & f \end{bmatrix}^T \tag{B.9}$$

$$\mathbf{S_k} = \begin{bmatrix} \ddots & \ddots & \ddots & & & & & & & & & & & & \\ & f & e & d & c & b & a & a & b & c & d & e & f & & \\ & & & & f & e & d & c & b & a & a & b & c & d & e & f \\ & & & & & & & & & & & & \ddots & \ddots & \ddots \end{bmatrix}^T \tag{B.10}$$

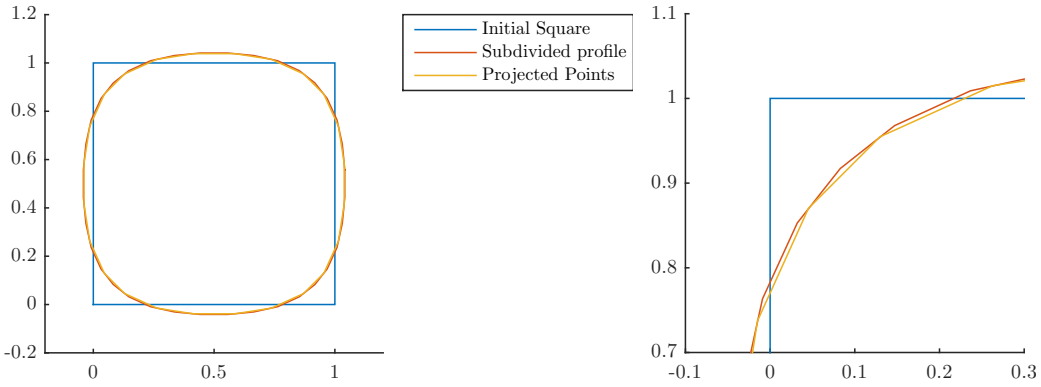
Where a through f coefficients that are solved for using Equation B.8. For a stable subdivision scheme the additional condition in Equation B.11 must hold. The vector of coefficients $\mathbf{m}$ is assembled into the subdivision matrix $\mathbf{S}_k$. This results in a set of simultaneous polynomial equations which can be set up and solved computationally. This process was carried out using the MATLAB symbolic toolbox. This allowed a wide range of masks and possible solutions to be explored rapidly. The coefficients that were selected for this work are presented in Equation B.12.

$$\text{For each row of } \mathbf{S_k}: \quad \sum_i^n s_{k,i} = 1 \tag{B.11}$$

$$\begin{aligned} a &= \frac{21}{20} - \frac{\sqrt{42}}{\sqrt{800}} & b &= \frac{21}{40} & c &= \frac{\sqrt{42}}{\sqrt{800}} - \frac{1}{20} \\ d &= 0 & e &= -\frac{1}{40} & f &= 0 \end{aligned} \tag{B.12}$$

B.3 Proposed Developments

While this appendix has shown the possibility to develop a subdivision which matches area, the chosen layout has limitations which make it unsuitable for aerodynamic use; namely it does not support sharp points, and it is oscillatory. Contrary to well established subdivision schemes, a rule for corners has not been developed. This rule is necessary to avoid undesirable rounding of the trailing edge by enforcing a sharp corner. The area matching rule presented above while stable and smooth, leads to a very noisy curvature of the profile which can be undesirable in aerodynamic applications.



(a) General view (b) Detailed view

Figure B.2: Square subdivided twice using an area conserving scheme

Those limitations meant that the constant area subdivision was not used in this work; however further exploration of parameter layouts and solution for each layout would certainly lead to additional layouts with more desirable properties. A systematic, automatic approach to the development, by quantifying desirable and undesirable properties, would enable the rapid development of a better constant area subdivision scheme; which could be used in re-sampling, but also in optimisation as part of the MLSO. A more thorough exploration of this tool, would also look at the locality of the area conservation, as currently the global area is conserved but nothing is known of the local properties.

APPENDIX C

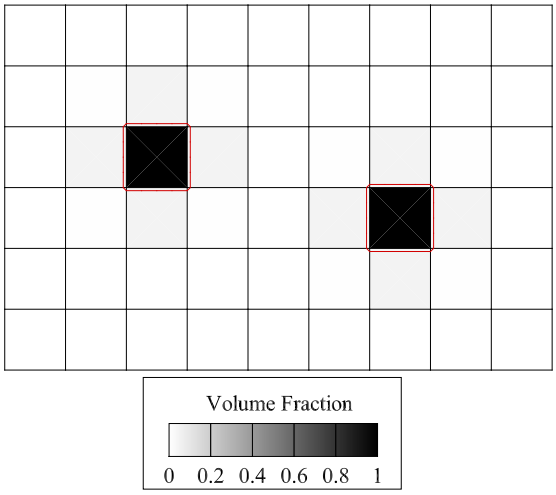
OPTIMISATION UNDER 'COMPONENT' CONSTRAINTS

The volume of solid (VOS) parameterisation method allows for the unique opportunity to specify regions of the design space which must be full. This gives the opportunity to give free reign to the optimisation method to design aerodynamic shells around bodies. This is achieved by specifying a minimum required volume fraction in some cells. The formulation of the optimisation problem is presented in Equation C.1, where $\mathbf{a}_c$ is a vector containing the minimum volume fraction in each design cell.

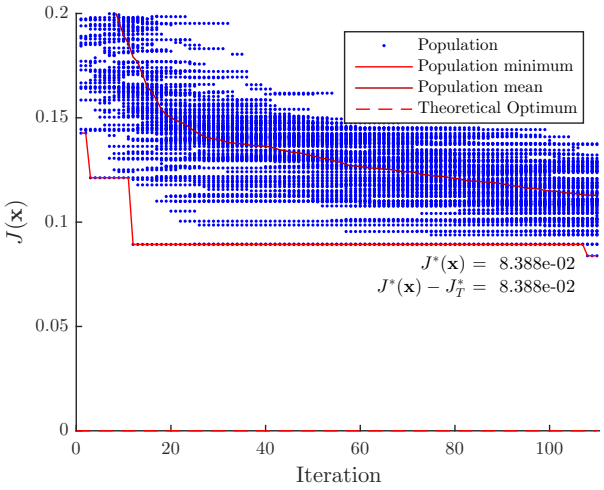
$$\begin{aligned} \min \quad & C_D \\ \text{s.t.} \quad & \mathbf{a} \geq \mathbf{a}_c \end{aligned} \tag{C.1}$$

Initial test were performed for two types of constraints referred to as *missile* and *smiley*. Both were performed with the DE optimiser. Attempts with a conjugate gradient optimiser invariably led to the convergence on local minima. Initial results are presented in Figure C.1 and C.2, for the missile and smiley respectively.

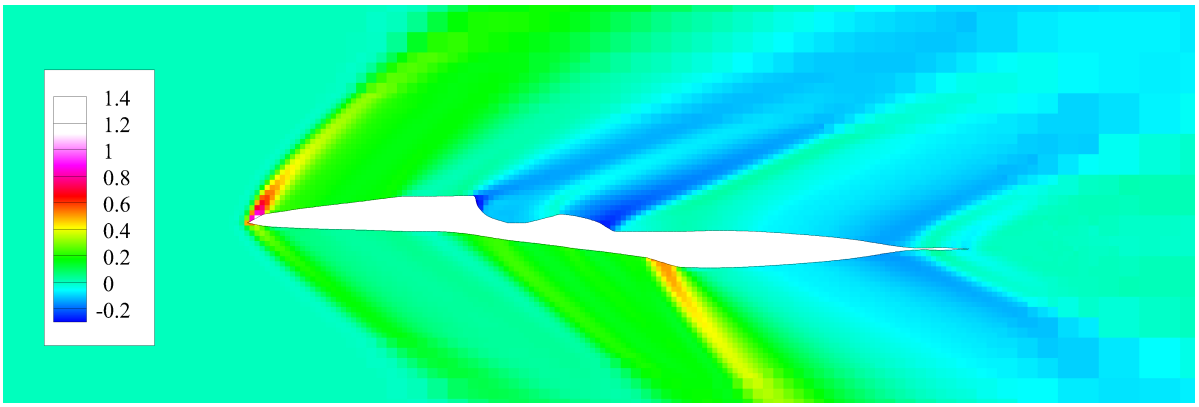
While the restricted snakes volume of solid (RSVS) can represent and explore these cases its reliance on differential evolution (DE) in these examples makes it inefficient. A further study of similar constraints with the integrated parameterisation framework would be much more revealing of the properties of the design and constraint spaces. These cases are also a precursor to the design of parts of a larger objects which have been suggested as an application of the RSVS and restricted surface volume of solid (3D-RSVS).



(a) Constraint values and minimum profile

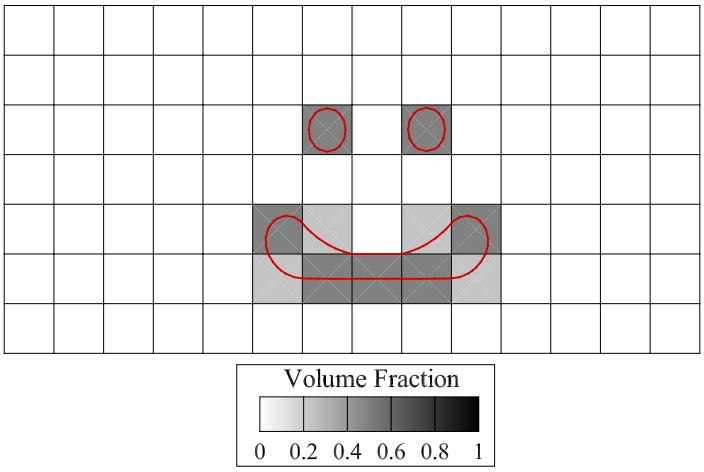


(b) Convergence history

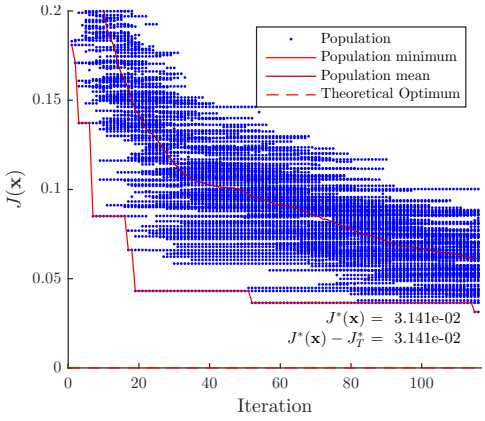


(c) Optimum profile

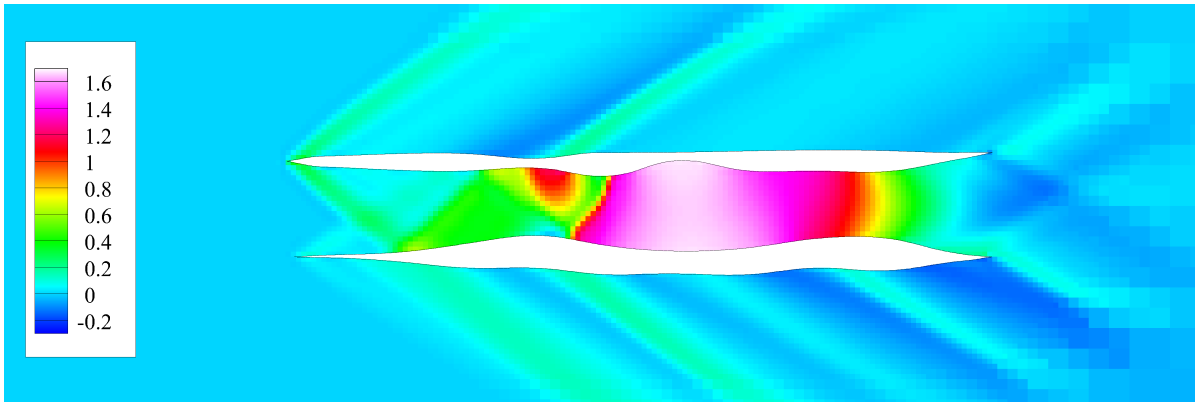
Figure C.1: Results for the DE optimisation under the *missile* constraint



(a) Constraint values and minimum profile



(b) Convergence history



(c) Optimum profile

Figure C.2: Results for the DE optimisation under the *smiley* constraint

APPENDIX D

DEVELOPMENT OF THE MULTI-LEVEL SUBDIVISION OPTIMISATION (MLSO) FRAMEWORK

This appendix presents the implementation details of the multi-level subdivision optimisation (MLSO) and the combined MLSO-restricted snakes volume of solid (RSVS) framework which was presented in Section 6.4. The text in this section are taken from the joint publications with Kedward:

- "Efficient Multi-Resolution Approaches for Exploration of External Aerodynamic Shape and Topology" [248];
- "Optimisation of Multi-Modal Aerodynamic Shape and Topology Problems" [249].

The implementation and description of the MLSO was the responsibility of Kedward, hence its presentation as an appendix to the work performed in Section 6.4. The author's contribution to this section was in the validation of MLSO developments for the combined framework; running, quantifying and plotting the impact of changes to the parameterisation. This contribution is in Section D.4.2.

This appendix presents first the implementation of the MLSO as originally developed by Masters et al. [176]. The modifications needed to ensure exact matching of the RSVS profiles needed for topology optimisation are presented in Section D.2. In order to ensure effective optimisation additional constraints were needed which are discussed in Section D.3. Finally Section D.4 presents the process used to validate that the MLSO method would be robust enough to tackle the wide variety of geometries which are generated by the RSVS.

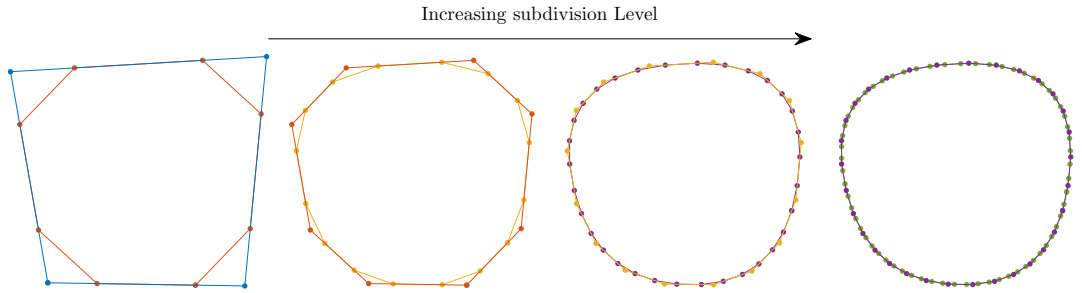


Figure D.1: Four levels of subdivision of a four point control polygon.

D.1 Efficient Shape Optimisation Using Multi-Resolution Subdivision Curves

A subdivision scheme defines a curve or surface as the limit of successive refinements starting from some initial polygon or polygonal mesh. Subdivision curves and surfaces currently dominate the entertainment graphics industry due to their unique topological flexibility compared to traditional spline-based methods, however the technology has seen growing attention in engineering applications [176, 238, 285, 286]. Recent work by Masters et al. applied multi-resolution subdivision curves in a hierarchical manner to parameterise aerofoil geometry and demonstrated improved efficiency and accuracy of aerodynamic shape optimisation [176]. Whereas the RSVS method provides complete topological flexibility which, in combination with a global search algorithm, also offers excellent coverage of the design space, the multilevel subdivision parameterisation represents an efficient and robust method for precisely resolving the local shape optimum for fixed topology configurations.

In their work, Masters et al. performed multiple optimisations sequentially, starting from a coarse control mesh and progressively refining; the effect of this is that shape control occurs at different length scales, starting with smooth large-scale changes and progressing to increasingly localised control. In this way high precision shape control can be performed without the deterioration in optimisation efficiency associated with localised shape parameterisation; when used in combination with an adjoint flow solver, providing surface sensitivities at greatly reduced cost, this results in significant reductions in computational cost. In addition, the subdivision method also inherently improves robustness against local optima since initial coarse control levels, which represent low-dimension approximations, allow the design space to be extensively explored early-on during optimisation.

The subdivision formulation is conceptually simple; given an initial control polygon

C_0, a refinement can be made linearly such that a new polygon is derived by a linear relationship using a subdivision matrix P:

$$C_1 = P_0 C_0 \tag{D.1}$$

This subdivision matrix encompasses two operations: a uniform topological refinement of the mesh (splitting) and a smoothing of the result (averaging), demonstrated in Figure D.1. Both operations are local and can hence be performed very efficiently. Subdivision schemes with unit maximum eigenvalue converge to a limit surface when applied ad infinitum; in practice the subdivision process can be truncated and the points of the final control polygon can be driven to their final limit positions by a limit matrix P_{eval}. Therefore the limit curve, sampled by the N^{th} subdivision level, can expressed in terms of the n^{th} level control polygon:

$$\phi_n = P_{eval} P_{N-1} P_{N-1} \ldots P_n \tag{D.2}$$

$$C_\infty = \phi_n C_n \tag{D.3}$$

Many subdivision schemes exist varying in the properties of the limit surfaces they generate and also, for subdivision surfaces, in the topology of the initial control mesh. Cubic B-Spline equivalent subdivision can be derived using the B-Spline knot-insertion property, this results in the following subdivision matrix for a two-dimensional curve:

$$P = \frac{1}{8}\begin{pmatrix} \ddots & \ddots & & & & & \\ \ddots & 1 & 6 & 1 & & & \\ & & 4 & 4 & & & \\ & & 1 & 6 & 1 & & \\ & & & 4 & 4 & & \\ & & & 1 & 6 & 1 & \ddots \\ & & & & & \ddots & \ddots \end{pmatrix} \tag{D.4}$$

When applied repeatedly, as in Equation D.2, the columns of the resulting ϕ matrix are cubic B-Spline basis functions.

A powerful extension is the multiresolution formulation, akin to the discrete wavelet transformation, which arises naturally since the subdivision surface definition is factorised as a sequence of refinement operations. By defining a coarsening operator R, in analogy to the inverse of the refinement matrix P of Equation D.4, then a multiresolution analysis can be performed by decomposing a fine shape representation into a coarse

approximation and a detail vector. The former is a coarser control level, which if refined again produces a smooth approximation to the input; the latter are the high frequency details lost during coarsening. Together the two outputs can be used to recover the original fine level input. A multilevel decomposition is performed by recursively applying the coarsening process resulting in a very coarse shape approximation and a set of detail vectors. It is this formulation that is used to preserve optimum geometry when performing progressive refinement optimisations and that allows any arbitrary input geometry to be represented by the subdivision formulation. The subdivision basis can be applied as a decomposition. An implicit approach, adopted by both Bandara [238] and Masters [9], is to minimise, in the least squares sense, the difference when the coarsened control points (obtained via R) are refined again using P. Which gives:

$$R_n = (P_n^T P_n)^{-1} P_n^T \tag{D.5}$$

Here R_n is a left pseudo-inverse, $R_n P_n = I$; i.e. refinement followed by coarsening, without editing, leads to the original set of control points; however, the same is not true for coarsening followed by refinement. Since exact transfer of shape information is required in both directions (coarsening and refining) a two-stage coarsening scheme is adopted; first the fine geometry is coarsened and then the information lost during coarsening is extracted and saved as a *detail* vector:

$$\begin{aligned} \mathbf{C}_n &= R_n \mathbf{C}_{n+1} \\ \mathbf{d}_n &= (I - P_n R_n)\mathbf{C}_{n+1} \end{aligned} \tag{D.6}$$

The first and second stages, which are not dependent on the output of each other, are analogous to a low-pass and high-pass filter respectively. The output of the first stage is a coarser control level, which if refined again produces a smooth approximation to the input. The output of the second stage are high frequency details that are lost during coarsening. Together the two outputs can be used to recover the original fine level data. A multilevel decomposition can now be performed by recursively applying the coarsening process of Equation D.6. A key result of this reverse-subdivision process is that it provides the framework to represent any shape using a set of multilevel subdivision surfaces. After decomposition, the surface geometry is derived from the n^{th} subdivision surface by the relationship:

$$\mathbf{C}_\infty = \phi_n \mathbf{C}_n + \sum_{i=n}^{N} \phi_{i+1} \mathbf{d}_i \tag{D.7}$$

It is this formulation that is used to preserve optimum geometry when performing progressive refinement optimisations and that allows any arbitrary input geometry to be represented by the subdivision formulation.

D.2 Exact Well-Posed Shape Control

A practical feature of the multiresolution subdivision representation is the inclusion of error terms to allow exact recovery of geometry. Common practice for spline-based shape parameterisation is to drop the error term since for an appropriate number of control points the approximation is usually sufficient. However in this work exact transfer of geometry between shape methods is highly desirable to avoid introducing ambiguity into the shape definition. In this work an alternate methodology for exact shape recovery is used whereby the error terms are included into the parameterisation as extra basis functions such that their amplitude can be modified as required by the search algorithm. This has been shown to overcome the adverse effects of including a constant error term into the shape definition [248]. Whereas the error terms are not 'good' basis functions in terms of orthogonality and smoothness, this is not of concern when used as part of a multilevel optimisation since the error components are progressively transferred to the standard subdivision basis when refinement is performed. This is particularly important since very low fidelity subdivision levels, with consequently larger error terms, will be used to take advantage of efficient exploration in low dimension design spaces. For these low fidelity subdivision curves, the error term can be of significant magnitude. Moreover, depending on the complexity of the shape, the error term may not have zero-mean, indicating that important shape features are essentially contained within the error term. Again, this is all the more important for this work where the RSVS method, in combination with a differential evolution optimiser, has the capability to generate a wide variety of shapes, for which simple spline approximations are not sufficient. In this section, an alternate methodology for including the error term is presented.

Repeated from Section 6.4.1.1 for clarity, the generation of geometry from subdivision is given by:

$$\mathbf{C}_\infty = \phi_n \mathbf{C}_n + \sum_{i=n}^{N} \phi_{i+1} \mathbf{d}_i$$

The first term in this equation represents the linear parameterisation where the shape design variables are derived from the stacked coordinates of the control polygon $\mathbf{C}_n$ at subdivision level n and the shape basis functions (cubic B-Spline equivalent) are contained within the matrix ϕ_n. The second term is a constant that when included allows recovery of the original shape via the accumulation of multiresolution error terms.

While enabling exact recovery of the initial shape, a disadvantage of the additional constant term is that it essentially represents a shape basis to which the optimisation algorithm has no access. Furthermore, since the error term results from a least-squares

process (reverse subdivision), it is necessarily not part of the parameterisation basis - the shape control to which the optimiser does have access. The result of this is that error terms of non-negligible magnitude 'offset' the shape parameterisation by a constant unmodifiable amount to produce a design space consisting entirely of oscillatory or 'peaky' shapes. This adversely affects the performance of shape optimisation.

Since as mentioned the error terms for this work cannot be guaranteed to be well-bounded, a modification to the linear parameterisation is used here whereby the error terms are included into the parameterisation as extra basis functions such that their amplitude can be modified as required by the search algorithm. Whereas the error terms are not 'good' basis functions in terms of orthogonality and smoothness, this is not of concern when used as part of a multilevel optimisation since the error components are progressively transferred to the standard subdivision basis when refinement is performed; this introduces better localised control and reduces the magnitude of the error term. The augmented linear parameterisation is therefore given by:

$$\begin{pmatrix} \mathbf{C}_{\infty}^{x} \\ \mathbf{C}_{\infty}^{y} \end{pmatrix} = \begin{pmatrix} \phi_n & \mathbf{0} & \mathbf{e}_n^x & \mathbf{0} \\ \mathbf{0} & \phi_n & \mathbf{0} & \mathbf{e}_n^y \end{pmatrix} \begin{pmatrix} \mathbf{C}_n^x \\ \mathbf{C}_n^y \\ \beta^x \\ \beta^y \end{pmatrix} \tag{D.8}$$

where

$$\begin{pmatrix} \mathbf{e}_n^x & \mathbf{e}_n^y \end{pmatrix} = \sum_{i=n}^{N} \phi_{i+1} \mathbf{d}_i \tag{D.9}$$

D.3 Shape Constraints

Topology optimisation can be viewed as encompassing and generalising shape optimisation, however the two processes have a fundamental difference; whereas the shape problem permits a local linearisation, and hence the use of local gradient-based methods, the topological design space does not and consequently requires global search methods. As presented in the previous sections, the two parameterisation methods chosen for this work have been shown to be highly effective in their respective areas: r-Snakes for flexible global topology control, and subdivision curves for efficient local shape control. A key difference between the two is in their shape representation methods and the implications for optimisation. Shapes produced by the RSVS method are contours recovered under a minimal length objective. As a result, shape connectivity is implicitly included into the formulation such that smooth continuous profiles are generated. Moreover the snake-marching method accounts for intersections and merging of separate

profiles. By contrast, subdivision curves, and all spline-based shape representations, are simple shape functions in parametric space and do not include any consideration of the underlying shape connectivity; this is to say that the resulting design space naturally contains non-physical shapes and invalid shapes (e.g. oscillatory, intersecting, inverted etc.), i.e. the problem is under-constrained. This means that the shape problem requires additional geometric constraints such that full advantage can be taken of the subdivision parameterisation for efficient local optimisation. Unlike typical academic problems where the initial geometry is known and geometric constraints can be specified manually, the constraint definitions here must be extremely versatile such that they can be applied in an automated fashion to the wide variety of shapes generated by the global topology search. Similarly a balance needs to be struck between sufficient constraint for well-posed search directions and sufficient feasible design space for local exploration. This is achieved in this work through linear constraints implementing move-limits, and a non-linear constraint for surface mesh validity.

- **Overall maximum chord:** the chord-wise extents of all bodies is bounded
- **Outer move limit:** shape displacements in the outward normal direction must be within a polygon defined around each loop by offsetting the loop convex hull
- **Inner move limit:** shape displacements in the inward normal direction must not cross the initial camber line of the loop.
- **Surface element size:** surface elements cannot shrink or grow beyond specified fractions of their initial size

The first three constraints can be implemented linearly which is highly desirable since existing local search methods (see Section D.2) allow exact constraint satisfaction at all iterations. The final constraint is non-linear which therefore requires differentiation to provide sensitivities and is not guaranteed to be satisfied at all iterations. The constraint function for element size is trivial and hence calculation of sensitivities, even by finite difference, does not impose any noticeable computational burden. Moreover, the highly sparse Jacobian means that the sensitivities can be efficiently evaluated when scaling to larger geometries. Similarly, the partial violation of the element size constraint during optimisation does not represent any realistic deficiency of the solution.

D.4 Configuration and Validation

The purpose of the hybrid optimisation framework is to improve the exploration of the topological design space allowed by the RSVS parameterisation. Inherent in the design of the hybrid method is the potential for a variety of different configurations in executing the optimisation. This includes settings such as parameterisation setup, number of iterations, population management, etc. The key challenge in choosing a configuration is to ensure that the local gradient-based optimisation runs reliably and effectively, regardless of input geometry and without human supervision.

Selection of a robust configuration involved the repeated testing of the combined framework with different settings on the same sample of starting geometries. During testing, focus was given to both the effectiveness of the method, to build confidence that the best solution found was close to the global optimum, as well as efficiency. This validation was performed using MS gradient-based on the starting population of a run of DE for an area constraint value of $c_A = 0.12$. The following key areas were identified for informal hypothesis-testing:

1. **Subdivision error basis:** does the novel error treatment presented here perform better during optimisation than existing methods?

2. **Efficient local optimisation:** what is a acceptable number of local iterations to perform at each multilevel optimisation and in what order should subdivision levels be used?

3. **Sufficient global optimisation:** what is an acceptable number of DE iterations to perform of global optimisation before starting local optimisation?

4. **Sub-population:** can the worst-performing profiles be discarded before starting local shape optimisation?

D.4.1 Treatment of the Subdivision Error Basis

In Section D.2 a qualitative argument is made for including the subdivision error into the shape design basis. In order to validate this *'basis'* methodology, its performance is benchmarked, in Figure D.2, against the two alternative procedures: inclusion of the error as a constant term (*'constant'*); and discarding the error term (*'removed')*. Figure D.2 shows statistical box plots summarising optimisation performance over the test population which has been performed at the first subdivision level for different error

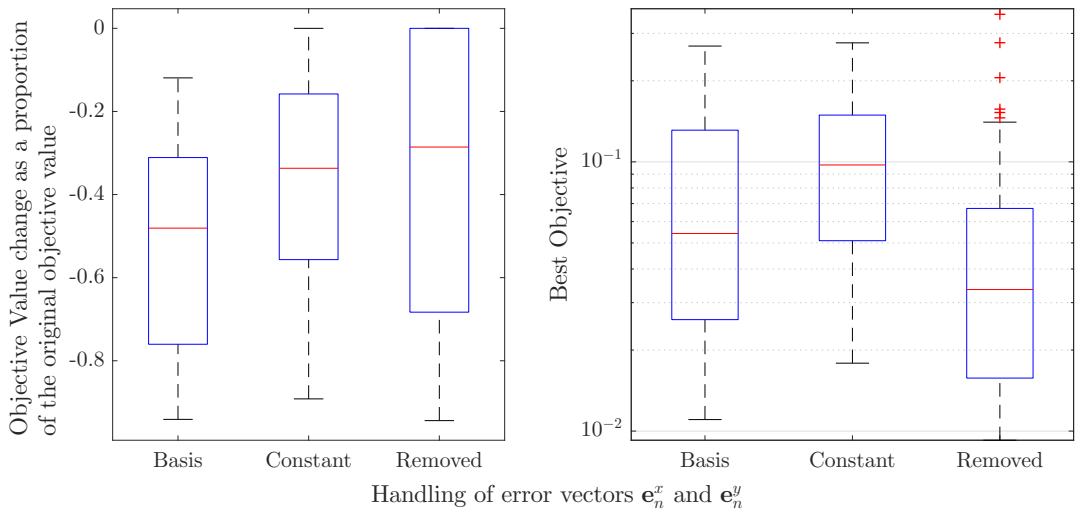


Figure D.2: Aggregate performance of single level validation runs broken down by subdivision configuration.

treatment methods. These tests were performed at the lowest resolution subdivision level because for lower subdivision levels, the error vector is larger, validating the three ways of handling it where the impact will be greatest.

It is clear that the 'basis' implementation achieves better overall improvements in objective value when compared to 'constant' and 'removed'. Moreover it is also statistically more consistent across the population. As shown by the plot of best objective values, the capability of the 'constant' implementation is noticeably limited compared to the other two which confirms the theory that the constant term has degraded the basis. Similarly, the 'removed' implementation does not perform as consistently across the population compared to 'basis' which shows that the optimiser is able to make good use of the additional error basis. Importantly the inclusion of the error term has not introduced untoward behaviour into the local shape optimisation, and based on this data it is decided to continue using this methodology.

D.4.2 Efficient Local Optimisation

The use of multi-resolution subdivision curves with progressively increasing fidelity contributes significantly to improved convergence of the local shape optimisation. In this work a straightforward progressive rule is adopted in which shape optimisations start at low fidelity and increase to high fidelity. Shown in Figure D.3 are multilevel results summarised across different subdivision starting levels. In general it is clear that starting at the lowest subdivision level is preferable in terms of overall result. This is

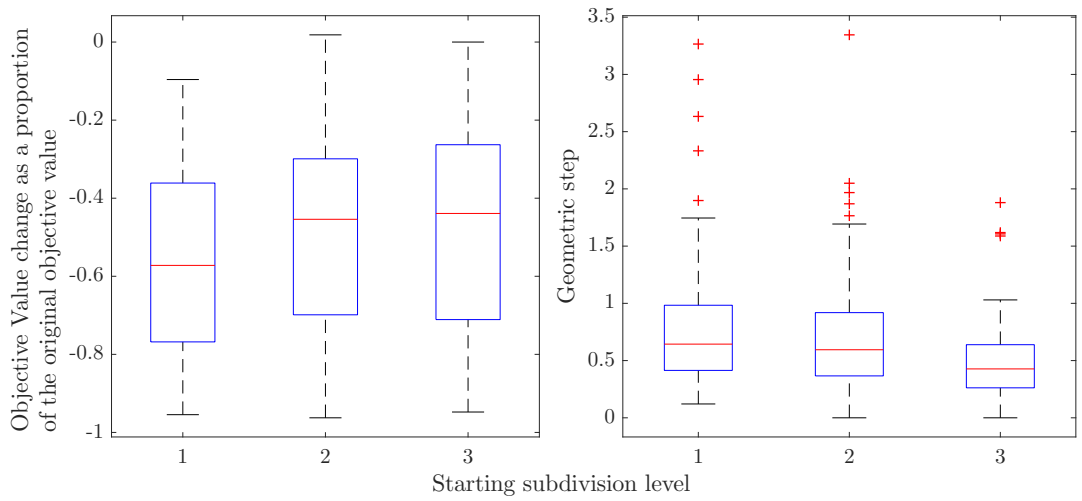


Figure D.3: Aggregate performance of multilevel validation runs broken down by subdivision level

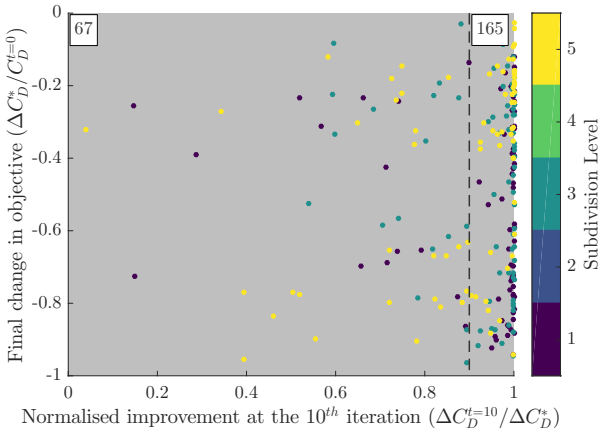


Figure D.4: Individual performance of single level validation runs at the tenth major iteration

intuitively understood by the fact that the lower fidelity control levels are able to take larger smoother steps than higher levels.

Convergence of each multilevel stage is fixed by using a constant number of major iterations for each level. This approach is adopted since numerical optimality of the local shape problem is insufficient in triggering a terminating condition for each level. Moreover, absolute convergence of intermediate levels is not required for the multilevel approach since higher levels offer more effective local shape control. A much larger number of iterations is specified for the final level.

For this work it is determined that ten major iterations for the intermediate levels is a good trade-off between computational cost and optimisation effectiveness. Figure

D.4 shows the individual performance of population members at the tenth iteration of a single level optimisation. Points are distributed along the x-axis by the ratio of the current objective improvement at the tenth iteration to their final improvement; points farther along the x-axis have reached a larger proportion of what they will eventually achieve. It is clear that by the tenth iteration the majority of the population have reached at least 90% of their capability. The points are distributed along the y-axis based on how well they will eventually perform compared to the initial objective value; points lower down the y-axes are the better individuals at the end of the shape optimisation.

The points of importance are those that have good potential (lower down the y-axis) but which have not yet reached 90% of their eventual performance; focussing on these points it is clear that they mostly belong to the fifth subdivision level. This high subdivision level offers high fidelity control (hence good potential) but with lower convergence rate. However the use of a multi-resolution approach here for the subdivision parameterisation avoids this slow convergence and hence these points are not of concern.

BIBLIOGRAPHY

[1] J. Slotnick, A. Khodadoust, J. Alonso, W. Gropp, and D. Mavriplis, "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," tech. rep., 2014.

[2] C. L. Navier, "Mémoire sur les lois du mouvement des fluides," *Mémoires de l'Académie des sciences de l'Institut de France*, vol. 6, pp. 389–416, 1823.

[3] G. G. Stokes, "On some cases of fluid motion," *Trans Cambridge Philos Soc*, vol. 8, pp. 105–165, 1843.

[4] T. 500, "The list." `https://www.top500.org/`. [Online; accessed 23/10/2019].

[5] Airbus, "Launch presentation of the A330-NEO." `https://www.ausbt.com.au/files/A330neo%20Launch%20Presentation.pdf`. [Online; accessed 13/06/2019].

[6] Altair, "A380 Weight Reduction through Optimization." `http://www.altairpd.com/CaseStudyDetail.aspx?id=10`. [Online; accessed 13/06/19].

[7] Altair, "Altair and Airbus APWorks Team up to Make Additive Manufacturing a Reality for Customers." `https://www.altair.com/NewsDetail.aspx?news_id=11269`. [Online; accessed 13/06/19].

[8] J. C. Vassberg and A. Jameson, "Industrial Applications of Aerodynamic Shape Optimization," tech. rep., Brussels, 2014.

[9] D. A. Masters, D. J. Poole, N. J. Taylor, T. C. S. Rendall, and C. B. Allen, "Influence of Shape Parameterization on a Benchmark Aerodynamic Optimization Problem," *Journal of Aircraft*, pp. 1–15, jun 2017.

[10] E. Hunter, "Alternate Detail Part Design and Analysis: Topology, Size, and Shape Optimization of CH-47 Chinook Underfloor Structure," in *62nd Annual Forum Proceedings-American Helicopter Society*, vol. 62, pp. 260–264, AMERICAN HELICOPTER SOCIETY, INC, 2006.

[11] J. D. Deaton and R. V. Grandhi, "A survey of structural and multidisciplinary continuum topology optimization: post 2000," *Struct Multidisc Optim*, vol. 49, pp. 1–38, 2014.

[12] M. P. Bendsøe and O. Sigmund, "Material interpolation schemes in topology optimization," *Archive of Applied Mechanics (Ingenieur Archiv)*, vol. 69, pp. 635–654, nov 1999.

[13] G. I. N. Rozvany, "A critical review of established methods of structural topology optimization," *Structural and Multidisciplinary Optimization*, vol. 37, pp. 217–237, jan 2009.

[14] N. Aage, E. Andreassen, B. S. Lazarov, and O. Sigmund, "Giga-voxel computational morphogenesis for structural design," *Nature*, vol. 550, pp. 84–86, oct 2017.

[15] Dassault Systemes, "What's New in SOLIDWORKS 2018: Topology Optimization of a Plate." `http://help.solidworks.com/2018/english/WhatsNew/c_topology_optimization_plate.htm`. [Online; accessed 2019-06-13].

[16] Autodesk, "Generative Design with Autodesk Nastran Topology Optimization." `https://www.autodesk.com/autodesk-university/class/Generative-Design-Autodesk-Nastran-Topology-Optimization-2016`. [Online; accessed 13/06/2019].

[17] Boeing, "737 MAX Gallery." `http://www.boeing.com/commercial/737max/#/design-highlights/max-efficiency/max-at-winglet/`. [Online; accessed 13/06/2019].

[18] G. Piola, "Ferrari formula 1 front wing." `https://www.formula1.com/content/fom-website/en/latest/technical/2015/9/ferrari-sf15-t---low-downforce-front-wing/_jcr_content/featureContent/manual_gallery/image1.img.2048.medium.jpg/1441192443879.jpg`. [Online; accessed 2017-04-05].

[19] C. Mattheck and S. Burkhardt, "A new method of structural shape optimization based on biological growth," *International Journal of Fatigue*, vol. 12, pp. 185–190, may 1990.

[20] G. Dantzing, "Linear Programming," in *Proceedings of Symposium on Modern Calculating Machinery and Numerical Methods*, (UCLA), 1948.

[21] E. W. Weisstein, "Least Squares Fitting.." http://mathworld.wolfram.com/LeastSquaresFitting.html. From MathWorld–A Wolfram Web Resource.

[22] E. Simoncelli, "Least squares optimization," *Lecture Notes, http://www.cns.nyu.edu/eero/teaching.html*, 2003.

[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, oct 1986.

[24] H. H. Rosenbrock, "An Automatic Method for Finding the Greatest or Least Value of a Function," *The Computer Journal*, vol. 3, pp. 175–184, mar 1960.

[25] SciPy, "SciPy optimize package." https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html#scipy.optimize.minimize. [Online; accessed 21/06/2019].

[26] J. Kiefer, "Sequential minimax search for a maximum," *Proceedings of the American Mathematical Society*, vol. 4, pp. 502–502, mar 1953.

[27] G. B. Dantzig, "A History of Scientific Computing," ch. Origins of, pp. 141–151, New York, NY, USA: ACM, 1990.

[28] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, vol. 116 of *International Series in Operations Research & Management Science*. Boston, MA: Springer US, 2008.

[29] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing - STOC '84*, (New York, New York, USA), pp. 302–311, ACM Press, 1984.

[30] J. A. Nelder and R. Mead, "A Simplex Method for Function Minimization," *The Computer Journal*, vol. 7, pp. 308–313, jan 1965.

[31] S. Singer and J. Nelder, "Nelder-Mead algorithm," *Scholarpedia*, vol. 4, no. 7, p. 2928, 2009.

[32] N. F. Foster and G. S. Dulikravich, "Three-Dimensional Aerodynamic Shape Optimization Using Genetic and Gradient Search Algorithms," *Journal of Spacecraft and Rockets*, vol. 34, pp. 36–42, jan 1997.

[33] A. Cauchy, "Méthode générale pour la résolution des systemes d'équations simultanées," *Comp. Rend. Sci. Paris*, vol. 25, no. 1847, pp. 536–538, 1847.

[34] H. B. Curry, "The method of steepest descent for non-linear minimization problems," *Quarterly of Applied Mathematics*, vol. 2, pp. 258–261, oct 1944.

[35] P. Wolfe, "Convergence Conditions for Ascent Methods," *SIAM Review*, vol. 11, pp. 226–235, apr 1969.

[36] P. Wolfe, "Convergence Conditions for Ascent Methods. II: Some Corrections," *SIAM Review*, vol. 13, pp. 185–188, apr 1971.

[37] a. Jameson, L. Martinelli, and N. Pierce, "Optimum Aerodynamic Design Using the Navier-Stokes Equations," *Theoretical and Computational Fluid Dynamics*, vol. 10, pp. 213–237, jan 1998.

[38] M. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, p. 409, dec 1952.

[39] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, feb 1964.

[40] E. Polak and G. Ribiere, "Note sur la convergence de méthodes de directions conjuguées," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 3, no. 16, pp. 35–43, 1969.

[41] Y. H. Dai and Y. Yuan, "A Nonlinear Conjugate Gradient Method with a Strong Global Convergence Property," *SIAM Journal on Optimization*, vol. 10, pp. 177–182, jan 1999.

[42] W. W. Hager and H. Zhang, "A Survey of Nonlinear Conjugate Gradient Methods," *Pacific Journal of Optimization*, vol. 2, no. 1, pp. 35–58, 2006.

[43] J. E. Peter and R. P. Dwight, "Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches," *Computers & Fluids*, vol. 39, pp. 373–391, mar 2010.

[44] D. P. Bertsekas, *Nonlinear Programming*. Belmont, Massachussets: Athena Scientific, 2nd editio ed., 1999.

[45] D. C. Liu and J. Nocedal, "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical Programming*, vol. 45, pp. 503–528, 1989.

[46] C. G. Broyden, J. E. Dennis, and J. J. Moré, "On the Local and Superlinear Convergence of Quasi-Newton Methods," *IMA Journal of Applied Mathematics*, vol. 12, no. 3, pp. 223–245, 1973.

[47] A. Forsgren, P. E. Gill, and M. H. Wright, "Interior Methods for Nonlinear Optimization," *SIAM Review*, vol. 44, no. 4, pp. 525–597, 2002.

[48] R. B. Wilson, *A simplical algorithm for concave programming*. Phd thesis, Harvard University, 1963.

[49] P. T. Boggs and J. W. Tolle, "Sequential Quadratic Programming," *Acta Numerica*, vol. 4, pp. 1–51, jan 1995.

[50] P. E. Gill, W. Murray, and M. a. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, vol. 12, pp. 979–1006, jan 2002.

[51] D. I. Papadimitriou and K. C. Giannakoglou, "Direct, adjoint and mixed approaches for the computation of Hessian in airfoil design problems," *International Journal for Numerical Methods in Fluids*, vol. 56, pp. 1929–1943, apr 2008.

[52] D. Shi-Dong and S. Nadarajah, "Approximate Hessian for accelerated convergence of aerodynamic shape optimization problems in an adjoint-based framework," *Computers and Fluids*, vol. 168, pp. 265–284, 2018.

[53] D. J. Poole, C. B. Allen, and T. C. S. Rendall, "Comparison of local and global constrained aerodynamic shape optimization," *32nd AIAA Applied Aerodynamics Conference 2014*, no. June, pp. 1–15, 2014.

[54] H. Gagnon and D. W. Zingg, "Two-Level Free-Form and Axial Deformation for Exploratory Aerodynamic Shape Optimization," *AIAA Journal*, vol. 53, pp. 2015–2026, jul 2015.

[55] E. Lee, K. A. James, and J. R. R. A. Martins, "Stress-constrained topology optimization with design-dependent loading," *Structural and Multidisciplinary Optimization*, vol. 46, no. 5, pp. 647–661, 2012.

[56] G. K. W. Kenway and J. R. R. A. Martins, "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, vol. 51, no. 1, pp. 144–160, 2014.

[57] D. A. Masters, N. J. Taylor, T. C. S. Rendall, C. B. Allen, and D. J. Poole, "Geometric Comparison of Aerofoil Shape Parameterization Methods," *AIAA Journal*, vol. 55, no. 5, pp. 1575–1589, 2017.

[58] O. Chernukhin and D. W. Zingg, "Multimodality and Global Optimization in Aerodynamic Design," *AIAA Journal*, vol. 51, pp. 1342–1354, jun 2013.

[59] W. Karush, *Minima of Functions of Several Variables with Inequalities as Side Constraints*. PhD thesis, Univ. of Chicago, Chicago, Illinois, 1939.

[60] H. W. Kuhn and A. W. Tucker, "Nonlinear Programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (J. Neyman, ed.), pp. 481—-492, University of California Press, 1951.

[61] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, IEEE, 1995.

[62] M. Khurana, H. Winarto, and A. Sinha, "Airfoil Optimisation by Swarm Algorithm with Mutation and Artificial Neural Networks," in *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, no. January, (Reston, Virigina), pp. 1–19, American Institute of Aeronautics and Astronautics, jan 2009.

[63] L. Blasi and G. D. Core, "Particle Swarm Approach in Finding Optimum Aircraft Configuration," *Journal of Aircraft*, vol. 44, pp. 679–683, mar 2007.

[64] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, pp. 28–39, nov 2006.

[65] A. H. Gandomi and A. H. Alavi, "Krill herd: A new bio-inspired optimization algorithm," *Communications in Nonlinear Science and Numerical Simulation*, vol. 17, pp. 4831–4845, dec 2012.

[66] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, mar 2014.

[67] H.-S. Wu and F.-M. Zhang, "Wolf Pack Algorithm for Unconstrained Global Optimization," *Mathematical Problems in Engineering*, vol. 2014, pp. 1–17, 2014.

[68] X.-S. Yang, "Metaheuristic Optimization," *Scholarpedia*, vol. 6, no. 8, p. 11472, 2011.

[69] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[70] T. L. Holst and T. H. Pulliam, "Evaluation of Genetic Algorithm Concepts Using Model Problems. Part 1: Single-Objective Optimization," tech. rep., NASA Ames Research Center; Moffett Field, CA, United States, Moffett Field, 2003.

[71] H. Bersini, M. Dorigo, S. Langerman, G. Seront, and L. Gambardella, "Results of the first international contest on evolutionary optimisation (1st ICEO)," in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 611–615, IEEE, 1996.

[72] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, pp. 1980–1987, IEEE, 2004.

[73] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[74] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 4–31, feb 2011.

[75] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, pp. 173–194, dec 2011.

[76] S. Das, S. S. Mullick, and P. Suganthan, "Recent advances in differential evolution – An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, apr 2016.

[77] D. J. Poole and C. B. Allen, "Constrained niching using differential evolution," *Swarm and Evolutionary Computation*, vol. 44, pp. 74–100, feb 2019.

[78] S. Y. Wang, K. Tai, and M. Y. Wang, "An enhanced genetic algorithm for structural topology optimization," *International Journal for Numerical Methods in Engineering*, vol. 65, pp. 18–44, jan 2006.

[79] D. Poole, C. Allen, and T. Rendall, "High-fidelity aerodynamic shape optimization using efficient orthogonal modal design variables with a constrained global optimizer," *Computers & Fluids*, vol. 143, pp. 1–15, 2017.

[80] O. Sigmund, "On the usefulness of non-gradient approaches in topology optimization," *Structural and Multidisciplinary Optimization*, vol. 43, no. 5, pp. 589–596, 2011.

[81] D. J. Poole, C. B. Allen, and T. C. S. Rendall, "Global Optimization of Wing Aerodynamic Optimization Case Exhibiting Multimodality," *Journal of Aircraft*, vol. 55, pp. 1576–1591, jul 2018.

[82] D. J. Poole, C. B. Allen, and T. Rendall, "Identifying Multiple Optima in Aerodynamic Design Spaces," in *2018 Multidisciplinary Analysis and Optimization Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jun 2018.

[83] J. He and X. Yao, "Average Drift Analysis and Population Scalability," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 1–1, 2016.

[84] L. M. Schmitt, "Theory of genetic algorithms," *Theoretical Computer Science*, vol. 259, pp. 1–61, may 2001.

[85] M. Dorigo and C. Blum, "Ant colony optimization theory: A survey," *Theoretical Computer Science*, vol. 344, no. 2-3, pp. 243–278, 2005.

[86] F. Van Den Bergh and A. P. Engelbrecht, "A convergence proof for the particle swarm optimiser," *Fundamenta Informaticae*, vol. 105, no. 4, pp. 341–374, 2010.

[87] S. Dasgupta, S. Das, A. Biswas, and A. Abraham, "On stability and convergence of the population-dynamics in differential evolution," *AI Communications*, vol. 22, pp. 1–20, jan 2009.

[88] S. Ghosh, S. Das, A. V. Vasilakos, and K. Suresh, "On Convergence of Differential Evolution Over a Class of Continuous Functions With Unique Global Optimum," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, pp. 107–124, feb 2012.

[89] A. V. Fiacco and Y. Ishizuka, "Sensitivity and stability analysis for nonlinear programming," *Annals of Operations Research*, vol. 27, pp. 215–235, dec 1990.

[90] C. Buskens and H. Maurer, "SQP-methods for solving optimal control problems with control and state constraints: Adjoint variables, sensitivity analysis and real-time control," *Journal of Computational and Applied Mathematics*, vol. 120, pp. 85–108, 2000.

[91] M. S. Phadke, *Quality Engineering Using Robust Design*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1st ed., 1995.

[92] G.-j. Park, T.-h. Lee, K. H. Lee, and K.-H. Hwang, "Robust Design: An Overview," *AIAA Journal*, vol. 44, pp. 181–191, jan 2006.

[93] C. Schillings and V. Schulz, "On the influence of robustness measures on shape optimization with stochastic uncertainties," *Optimization and Engineering*, vol. 16, pp. 347–386, jun 2015.

[94] F. Fusi, A. Guardone, G. Quaranta, and P. M. Congedo, "Multifidelity Physics-Based Method for Robust Optimization Applied to a Hovering Rotor Airfoil," *AIAA Journal*, vol. 53, pp. 3448–3465, nov 2015.

[95] G. K. W. Kenway and J. R. R. A. Martins, "Multipoint Aerodynamic Shape Optimization Investigations of the Common Research Model Wing," *AIAA Journal*, vol. 54, pp. 113–128, jan 2016.

[96] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking Multiple Solutions: An Updated Survey on Niching Methods and Their Applications," *IEEE Transactions on Evolutionary Computation*, vol. 21, pp. 518–538, aug 2017.

[97] K. Deb, "Multi-objective Optimization," in *Search Methodologies*, pp. 403–449, Boston, MA: Springer US, 2014.

[98] R. Dufo-López, J. L. Bernal-Agustín, J. M. Yusta-Loyo, J. A. Domínguez-Navarro, I. J. Ramírez-Rosado, J. Lujano, and I. Aso, "Multi-objective optimization minimizing cost and life cycle emissions of stand-alone PV–wind–diesel systems with batteries storage," *Applied Energy*, vol. 88, pp. 4033–4041, nov 2011.

[99] T. Simpson, V. Toropov, V. Balabanov, and F. Viana, "Design and Analysis of Computer Experiments in Multidisciplinary Design Optimization: A Review of How Far

We Have Come - Or Not," in *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, (Reston, Virigina), American Institute of Aeronautics and Astronautics, sep 2008.

[100] G. Matheron, "Principles of geostatistics," *Economic Geology*, vol. 58, pp. 1246–1266, dec 1963.

[101] G. Matheron, "A Simple Substitute for Conditional Expectation : The Disjunctive Kriging," in *Advanced Geostatistics in the Mining Industry*, pp. 221–236, Dordrecht: Springer Netherlands, 1976.

[102] Z. Han and S. Görtz, "Hierarchical Kriging Model for Variable-Fidelity Surrogate Modeling," *AIAA Journal*, vol. 50, no. 9, pp. 1885–1896, 2012.

[103] T. J. Mackman, C. B. Allen, M. Ghoreyshi, and K. J. Badcock, "Comparison of Adaptive Sampling Methods for Generation of Surrogate Aerodynamic Models," *AIAA Journal*, vol. 51, pp. 797–808, apr 2013.

[104] P. G. Constantine, E. Dow, and Q. Wang, "Active Subspace Methods in Theory and Practice: Applications to Kriging Surfaces," *SIAM Journal on Scientific Computing*, vol. 36, no. 4, pp. A1500–A1524, 2014.

[105] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary optimization of computationally expensive problems via surrogate modeling," *AIAA Journal*, vol. 41, no. 4, pp. 687–696, 2003.

[106] J. Li, M. A. Bouhlel, and J. R. R. A. Martins, "Data-Based Approach for Fast Airfoil Analysis and Optimization," *AIAA Journal*, pp. 1–16, nov 2018.

[107] S. Koziel and L. Leifsson, "Efficient knowledge-based optimization of expensive computational models using adaptive response correction," *Journal of Computational Science*, vol. 11, pp. 1–11, nov 2015.

[108] D. Cinquegrana and E. Iuliano, "Investigation of adaptive design variables bounds in dimensionality reduction for aerodynamic shape optimization," *Computers & Fluids*, vol. 174, pp. 89–109, sep 2018.

[109] J. K. Pugh, L. B. Soros, and K. O. Stanley, "An Extended Study of Quality Diversity Algorithms," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO '16 Companion*, (New York, New York, USA), pp. 19–20, ACM Press, 2016.

[110] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret, "Robots that can adapt like animals," *Nature*, vol. 521, pp. 503–507, may 2015.

[111] A. Gaier, A. Asteroth, and J.-b. Mouret, "Aerodynamic Design Exploration through Surrogate-Assisted Illumination," in *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, AIAA AVIATION Forum, (AIAA 2017-3330)*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jun 2017.

[112] A. Gaier, A. Asteroth, and J.-B. Mouret, "Data-Efficient Design Exploration through Surrogate-Assisted Illumination," *Evolutionary Computation*, vol. 26, pp. 381–410, sep 2018.

[113] A. M. Morris, C. B. Allen, and T. C. S. Rendall, "High-fidelity aerodynamic shape optimization of modern transport wing using efficient hierarchical parameterization," *International Journal for Numerical Methods in Fluids*, vol. 63, pp. 297–312, may 2010.

[114] I. Kroo, S. Altus, R. Braun, P. Gage, and I. Sobieski, "Multidisciplinary optimization methods for aircraft preliminary design," in *5th Symposium on Multidisciplinary Analysis and Optimization*, (Reston, Virigina), American Institute of Aeronautics and Astronautics, sep 1994.

[115] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso, "The complex-step derivative approximation," *ACM Transactions on Mathematical Software*, vol. 29, pp. 245–262, sep 2003.

[116] J. R. R. A. Martins and A. B. Lambe, "Multidisciplinary Design Optimization: A Survey of Architectures," *AIAA Journal*, vol. 51, pp. 2049–2075, sep 2013.

[117] T. R. Brooks, G. K. W. Kenway, and J. R. R. A. Martins, "Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings," *AIAA Journal*, vol. 56, pp. 2840–2855, jul 2018.

[118] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, vol. 59, pp. 1075–1104, apr 2019.

[119] J. R. R. A. Martins and J. T. Hwang, "Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models," *AIAA Journal*, vol. 51, pp. 2582–2599, nov 2013.

[120] M. P. Bendsøe, "Optimal shape design as a material distribution problem," *Structural Optimization*, vol. 1, pp. 193–202, dec 1989.

[121] B. Hassani and E. Hinton, "A review of homogenization and topology optimization I—homogenization theory for media with periodic structure," *Computers & Structures*, vol. 69, no. 6, pp. 707–717, 1998.

[122] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, pp. 197–224, nov 1988.

[123] E. Oktay, H. U. Akay, and O. T. Sehitoglu, "Three-dimensional structural topology optimization of aerial vehicles under aerodynamic loads," *Computers & Fluids*, vol. 92, pp. 225–232, mar 2014.

[124] A. D. Taylor, A. D. Payot, C. B. Allen, and T. Rendall, "Structural Topology Optimisation with R-Snakes Volume of Solid," in *AIAA Scitech 2019 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2019.

[125] O. Sigmund, "A 99 line topology optimization code written in matlab," *Structural and Multidisciplinary Optimization*, vol. 21, pp. 120–127, Apr 2001.

[126] Y. Xie and G. Steven, "A simple evolutionary procedure for structural optimization," *Computers & Structures*, vol. 49, pp. 885–896, dec 1993.

[127] X. Huang and Y. M. Xie, "Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials," *Comput Mech*, vol. 43, pp. 393–401, 2009.

[128] P. Tanskanen, "The evolutionary structural optimization method: theoretical aspects," *Comput. Methods Appl. Mech. Engrg*, vol. 191, pp. 5485–5498, 2002.

[129] O. Sigmund and J. Petersson, "Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima," *Structural Optimization*, vol. 16, pp. 68–75, aug 1998.

[130] M. Y. Wang, X. Wang, and D. Guo, "A level set method for structural topology optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 192, pp. 227–246, jan 2003.

[131] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen, "Level-set methods for structural topology optimization: a review," *Structural and Multidisciplinary Optimization*, vol. 48, pp. 437–472, sep 2013.

[132] Z. Luo, N. Zhang, W. Gao, and H. Ma, "Structural shape and topology optimization using a meshless Galerkin level set method," *International Journal for Numerical Methods in Engineering*, vol. 90, pp. 369–389, apr 2012.

[133] M. P. Bendsøe and O. Sigmund, *Topology optimization: theory, methods, and applications*, vol. 2nd Editio. 2003.

[134] J. Sokolowski and A. Zochowski, "On the Topological Derivative in Shape Optimization," *SIAM Journal on Control and Optimization*, vol. 37, pp. 1251–1272, jan 1999.

[135] S. Amstutz, "The topological asymptotic for the Navier-Stokes equations," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 11, pp. 401–425, jul 2005.

[136] J. M. Mcdonough, "Lectures on Turbulence Physics, Mathematics and Modeling," tech. rep., 2007.

[137] J. F. Wendt, ed., *Computational Fluid Dynamics*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[138] T. C. S. Rendall and C. B. Allen, "Lecture notes: Numerical/potential flow methods and boundary layers," tech. rep., University of Bristol, Bristol, 2016.

[139] K. V. Belyaev, A. V. Garbaruk, M. L. Shur, M. K. H. Strelets, and P. R. Spalart, "Experience of Direct Numerical Simulation of Turbulence on Supercomputers," pp. 67–77, Springer, Cham, 2016.

[140] A. Lozano-Durán, M. J. P. Hack, and P. Moin, "Modeling boundary-layer transition in direct and large-eddy simulations using parabolized stability equations," *Physical Review Fluids*, vol. 3, p. 23901, 2018.

[141] J. Hall, T. C. S. Rendall, and C. B. Allen, "Optimisation of a Coastal Defence Geometry Using SPH," in *Proceedings of the 10th SPHERIC Workshop*, pp. 229–236, 2015.

[142] F. H. Harlow and F. H, "The particle-in-cell method for numerical solution of problems in fluid dynamics," tech. rep., Los Alamos National Laboratory (LANL), Los Alamos, NM (United States), mar 1962.

[143] R. A. Gentry, R. E. Martin, and B. J. Daly, "An Eulerian differencing method for unsteady compressible flow problems," *Journal of Computational Physics*, vol. 1, pp. 87–118, aug 1966.

[144] J. Brackbill and H. Ruppel, "FLIP: A method for adaptively zoned, particle-in-cell calculations of fluid flows in two dimensions," *Journal of Computational Physics*, vol. 65, pp. 314–343, aug 1986.

[145] S. Jakobsson and O. Amoignon, "Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization," *Computers & Fluids*, vol. 36, pp. 1119–1136, jul 2007.

[146] T. C. S. Rendall and C. B. Allen, "Unified fluid–structure interpolation and mesh motion using radial basis functions," *International Journal for Numerical Methods in Engineering*, vol. 74, pp. 1519–1559, jun 2008.

[147] C. S. Peskin, "The immersed boundary method," *Acta Numerica*, vol. 11, pp. 479–517, jan 2002.

[148] C. Zhu, J.-H. Seo, and R. Mittal, "A graph-partitioned sharp-interface immersed boundary solver for efficient solution of internal flows," *Journal of Computational Physics*, vol. 386, pp. 37–46, jun 2019.

[149] S. Peron, T. Renaud, I. Mary, C. Benoit, and M. Terracol, "An Immersed Boundary Method for preliminary design aerodynamic studies of complex configurations," in *23rd AIAA Computational Fluid Dynamics Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jun 2017.

[150] T. Renaud, C. Benoit, S. Peron, I. Mary, and N. Alferez, "Validation of an immersed boundary method for compressible flows," in *AIAA Scitech 2019 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2019.

[151] B. König and E. Fares, "Exa PowerFLOW Simulations for the Sixth AIAA Drag Prediction Workshop," *Journal of Aircraft*, vol. 55, pp. 1482–1490, jul 2018.

[152] T. Rendall and C. Allen, "CFD Simulation of Arbitrary Motion in Two-Dimensional Spacetime Using Cut-Cell Meshes," in *28th AIAA Applied Aerodynamics Conference*, no. AIAA 2010-4696, (Reston, Virigina), pp. 1–11, American Institute of Aeronautics and Astronautics, jun 2010.

[153] M. J. Aftosmis, M. J. Berger, and J. E. Melton, "Robust and efficient Cartesian mesh generation for component-based geometry," *AIAA Journal*, vol. 36, pp. 952–960, jan 1998.

[154] G. R. Anderson, M. Nemec, and M. J. Aftosmis, "Aerodynamic Shape Optimization Benchmarks with Error Control and Automatic Parameterization," in *53rd AIAA Aerospace Sciences Meeting*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[155] J. Hall, T. Rendall, C. Allen, and D. Poole, "A volumetric geometry and topology parameterisation for fluids-based optimisation," *Computers & Fluids*, vol. 148, pp. 137–156, apr 2017.

[156] J. Martins, P. Sturdza, and J. Alonso, "The connection between the complex-step derivative approximation and algorithmic differentiation," in *39th Aerospace Sciences Meeting and Exhibit*, vol. 921, (Reston, Virigina), p. 2001, American Institute of Aeronautics and Astronautics, jan 2001.

[157] D. Koo and D. W. Zingg, "Progress in Aerodynamic Shape Optimization Based on the Reynolds-Averaged Navier-Stokes Equations," in *54th AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2016.

[158] G. K. Kenway, D. A. Burdette, and J. Martins, "Multipoint Aerodynamic Shape Optimization Investigations of the Common Research Model Wing," in *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[159] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, no. 3, pp. 233–260, 1988.

[160] S. K. Nadarajah and A. Jameson, "Optimum Shape Design for Unsteady Three-Dimensional Viscous Flows Using a Nonlinear Frequency-Domain Method," *Journal of Aircraft*, vol. 44, no. 5, pp. 1513–1527, 2007.

[161] M. B. Giles and N. A. Pierce, "An Introduction to the Adjoint Approach to Design," *Flow, Turbulence and Combustion*, vol. 65, no. 3/4, pp. 393–415, 2000.

[162] S. Nadarajah and A. Jameson, "A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization," in *38th Aerospace Sciences Meeting and Exhibit*, (Reston, Virigina), American Institute of Aeronautics and Astronautics, jan 2000.

[163] J. C. Vassberg, N. A. Harrison, D. L. Roman, and A. Jameson, "A Systematic Study on the Impact of Dimensionality for a Two-Dimensional Aerodynamic Optimization Model Problem," in *29th AIAA Applied Aerodynamics Conference*, no. 3176, (Honolulu, Hawaii), pp. 1–19, 2011.

[164] P. Castonguay and S. Nadarajah, "Effect of Shape Parameterization on Aerodynamic Shape Optimization," in *45th AIAA Aerospace Sciences Meeting and Exhibit*, no. January, (Reston, Virigina), pp. 1–20, American Institute of Aeronautics and Astronautics, jan 2007.

[165] B. Kulfan and J. Bussoletti, "'Fundamental" Parameteric Geometry Representations for Aircraft Component Shapes," in *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, vol. 1, (Reston, Virigina), pp. 547–591, American Institute of Aeronautics and Astronautics, sep 2006.

[166] H. Sobieczky, "Geometry Generator for CFD and Applied Aerodynamics," in *New Design Concepts for High Speed Air Transport*, pp. 137–157, Vienna: Springer Vienna, 1997.

[167] H. Sobieczky, "Parametric Airfoils and Wings," in *Notes on Numerical Fluid Mechanics vol. 65* (K. Fujii and G. S. Dulikravich, eds.), ch. Parametric, pp. 71–87, Wiesbaden: Vieweg+Teubner Verlag, 1998.

[168] F. Bisson and S. Nadarajah, "Adjoint-Based Aerodynamic Optimization Framework," in *52nd Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2014.

[169] M. Gariepy, J.-Y. Trepanier, E. Petro, B. Malouin, C. Audet, S. LeDigabel, and C. Tribes, "Direct Search Airfoil Optimization Using Far-Field Drag Decomposition Results," in *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[170] R. M. Hicks and P. A. Henne, "Wing Design by Numerical Optimization," *Journal of Aircraft*, vol. 15, pp. 407–412, jul 1978.

[171] D. Chauhan, P. Chandrashekarappa, and R. Duvigneau, "Wing shape optimization using FFD and twist parameterization," in *12th Aerospace Society of India CFD Symposium*, (Bangalore), 2010.

[172] C. B. Allen and T. C. S. Rendall, "CFD-based optimization of hovering rotors using radial basis functions for shape parameterization and mesh deformation," *Optimization and Engineering*, vol. 14, pp. 97–118, mar 2013.

[173] D. J. J. Toal, N. W. Bressloff, A. J. Keane, and C. M. E. Holden, "Geometric Filtration Using Proper Orthogonal Decomposition for Aerodynamic Design Optimization," *AIAA Journal*, vol. 48, no. 5, pp. 916–928, 2010.

[174] D. J. Poole, C. B. Allen, and T. C. S. Rendall, "Metric-Based Mathematical Derivation of Efficient Airfoil Design Variables," *AIAA Journal*, vol. 53, pp. 1349–1361, may 2015.

[175] G. R. Anderson and M. J. Aftosmis, "Adaptive Shape Control for Aerodynamic Design," in *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[176] D. A. Masters, N. J. Taylor, T. C. S. Rendall, and C. B. Allen, "Multilevel Subdivision Parameterization Scheme for Aerodynamic Shape Optimization," *AIAA Journal*, vol. 55, pp. 3288–3303, oct 2017.

[177] S. H. Berguin and D. N. Mavris, "Dimensionality Reduction Using Principal Component Analysis Applied to the Gradient," *AIAA Journal*, vol. 53, no. 4, pp. 1078–1090, 2015.

[178] I. Prilepov, H. Obermaier, E. Deines, C. Garth, and K. I. Joy, "Cubic gradient-based material interfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 10, pp. 1687–1699, 2013.

[179] S. K. Nadarajah, "Aerodynamic Design Optimization: Drag Minimization of the NACA 0012 in Transonic Inviscid Flow," 2013.

[180] S. K. Nadarajah, "Aerodynamic Design Optimization: Drag Minimization of the RAE 2822 in Transonic Viscous Flow," 2013.

[181] L. Osusky, H. Buckley, and D. W. Zingg, "Aerodynamic Design Optimization Discussion Group Case 4 : Single- and multi-point optimization problems based on the CRM wing Information Common to All Problems."

[182] G. K. W. Kenway and J. R. R. A. Martins, "Case 5 : CRM Wing-Body-Tail Optimization at Flight Reynolds Number," 2015.

[183] E. Fabiano and D. J. Mavriplis, "Adjoint-Based Aerodynamic Design On Unstructured Meshes," in *54th AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2016.

[184] F. Fusi, G. Quaranta, A. Guardone, and P. M. Congedo, "Drag minimization of an isolated airfoil in transonic inviscid flow by means of genetic algorithms," in *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[185] E. Iuliano and D. Quagliarella, "Evolutionary Optimization of Benchmark Aerodynamic Cases using Physics-based Surrogate Models," in *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[186] C. Lee, D. Koo, K. Telidetzki, H. Buckley, H. Gagnon, and D. W. Zingg, "Aerodynamic Shape Optimization of Benchmark Problems Using Jetstream," in *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[187] L. T. Leifsson, S. Koziel, Y. A. Tesfahunegn, S. Hosder, and J.-R. Gramanzini, "Application of Physics-Based Surrogate Models to Benchmark Aerodynamic Shape Optimization Problems," in *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[188] M. Meheut, D. Destarac, S. Ben Khelil, G. Carrier, A. Dumont, and J. Peter, "Gradient-Based Single and Multi-points Aerodynamic Optimizations with the

elsA Software," in *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[189] D. J. Poole, C. B. Allen, and T. Rendall, "Control Point-Based Aerodynamic Shape Optimization Applied to AIAA ADODG Test Cases," in *53rd AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2015-1947)*, no. January, (Reston, Virginia), pp. 1–20, American Institute of Aeronautics and Astronautics, jan 2015.

[190] J. Ren, A. S. Thelen, A. Amrit, X. Du, L. T. Leifsson, Y. Tesfahunegn, and S. Koziel, "Application of Multifidelity Optimization Techniques to Benchmark Aerodynamic Design Problems," in *54th AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2016.

[191] Y. Zhang, Z.-H. Han, L. Shi, and W.-P. Song, "Multi-round Surrogate-based Optimization for Benchmark Aerodynamic Design Problems," in *54th AIAA Aerospace Sciences Meeting*, no. 127, (Reston, Virginia), pp. 1–17, American Institute of Aeronautics and Astronautics, jan 2016.

[192] L. Kedward, C. B. Allen, and T. Rendall, "Gradient-Limiting Shape Control for Efficient Aerodynamic Optimisation," in *2018 Applied Aerodynamics Conference*, (Reston, Virginia), pp. 1–22, American Institute of Aeronautics and Astronautics, jun 2018.

[193] L. J. Kedward, C. B. Allen, and T. Rendall, "Gradient-Limiting Shape Optimisation Applied to AIAA ADODG Test Cases," in *AIAA Scitech 2019 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2019.

[194] D. Destarac, G. Carrier, G. R. Anderson, S. Nadarajah, D. J. Poole, J. C. Vassberg, and D. W. Zingg, "Example of a Pitfall in Aerodynamic Shape Optimization," *AIAA Journal*, vol. 56, pp. 1532–1540, apr 2018.

[195] J. E. Hicken, "Aerodynamic Design Optimization Workshop : Twist Optimization Case," 2013.

[196] A. W. Rizzi, M. Zhang, F. Bisson, S. Nadarajah, and J. B. Vos, "Comparitive Study of Two Optimization Frameworks Applied to Case III: Induced-Drag Minimization," in *54th AIAA Aerospace Sciences Meeting*, no. January, pp. 1–21, 2016.

[197] M. Meheut, A. Dumont, G. Carrier, and J. E. Peter, "Gradient-Based Optimization of CRM Wing-alone and Wing-body-tail Configurations by RANS Adjoint Technique," in *54th AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2016.

[198] S. Chen, Z. Lyu, G. K. Kenway, and J. Martins, "Aerodynamic Shape Optimization of the Common Research Model Wing-Body-Tail Configuration," in *53rd AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2015.

[199] G. K. Kenway and J. Martins, "Aerodynamic Shape Optimization of the CRM Configuration Including Buffet-Onset Conditions," in *54th AIAA Aerospace Sciences Meeting*, AIAA SciTech, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2016.

[200] N. Bons, X. He, C. A. Mader, and J. Martins, "Multimodality in Aerodynamic Wing Design Optimization," in *35th AIAA Applied Aerodynamics Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jun 2017.

[201] G. M. Streuber and D. W. Zingg, "Investigation of Multimodality in Aerodynamic Shape Optimization Based on the Reynolds Averaged Navier-Stokes Equations," in *35th AIAA Applied Aerodynamics Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jun 2017.

[202] X. Du, A. Amrit, A. S. Thelen, L. T. Leifsson, Y. Zhang, Z.-H. Han, and S. Koziel, "Aerodynamic Design of a Rectangular Wing in Subsonic Inviscid Flow by Direct and Surrogate-based Optimization," in *35th AIAA Applied Aerodynamics Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jun 2017.

[203] C. B. Dilgen, S. B. Dilgen, D. R. Fuhrman, O. Sigmund, and B. S. Lazarov, "Topology optimization of turbulent flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 331, pp. 363–393, apr 2018.

[204] N. Aage, T. H. Poulsen, A. Gersborg-Hansen, and O. Sigmund, "Topology optimization of large scale stokes flow problems," *Structural and Multidisciplinary Optimization*, vol. 35, pp. 175–180, feb 2008.

[205] S. Kreissl, G. Pingen, A. Evgrafov, and K. Maute, "Topology optimization of flexible micro-fluidic devices," *Structural and Multidisciplinary Optimization*, vol. 42, pp. 495–516, oct 2010.

[206] E. Burman and P. Hansbo, "Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method," *Applied Numerical Mathematics*, vol. 62, pp. 328–341, apr 2012.

[207] Y. Deng, Z. Liu, P. Zhang, Y. Liu, and Y. Wu, "Topology optimization of unsteady incompressible Navier–Stokes flows," *Journal of Computational Physics*, vol. 230, pp. 6688–6708, jul 2011.

[208] C. chen, K. Yaji, T. Yamada, K. Izui, and S. Nishiwaki, "Local-in-time adjoint-based topology optimization of unsteady fluid flows using the lattice Boltzmann method," *Mechanical Engineering Journal*, vol. 4, no. 3, pp. 17–00120–17–00120, 2017.

[209] F. Dugast, Y. Favennec, C. Josset, Y. Fan, and L. Luo, "Topology optimization of thermal fluid flows with an adjoint Lattice Boltzmann Method," *Journal of Computational Physics*, vol. 365, pp. 376–404, 2018.

[210] G. H. Yoon, "Topology optimization for turbulent flow with Spalart–Allmaras model," *Computer Methods in Applied Mechanics and Engineering*, vol. 303, pp. 288–311, may 2016.

[211] C. H. Villanueva and K. Maute, "CutFEM topology optimization of 3D laminar incompressible flow problems," *Computer Methods in Applied Mechanics and Engineering*, vol. 320, pp. 444–473, 2017.

[212] T. Borrvall and J. Petersson, "Topology optimization of fluids in Stokes flow," *International Journal for Numerical Methods in Fluids*, vol. 41, pp. 77–107, jan 2003.

[213] A. Payot, T. Rendall, and C. Allen, "Restricted snakes volume of solid (RSVS): A parameterisation method for topology optimisation of external aerodynamics," *Computers & Fluids*, vol. 182, pp. 60–84, mar 2019.

[214] J. Hall, T. C. S. Rendall, and C. B. Allen, "Optimisation using smoothed particle hydrodynamics with volume-based geometry control," *Structural and Multidisciplinary Optimization*, 2017.

[215] T. McInerney and D. Terzopoulos, "T-snakes: Topology adaptive snakes," *Medical Image Analysis*, vol. 4, pp. 73–91, jun 2000.

[216] R. Delgado-Gonzalo, V. Uhlmann, D. Schmitter, and M. Unser, "Snakes on a Plane: A perfect snap for bioimage analysis," *IEEE Signal Processing Magazine*, vol. 32, pp. 41–48, jan 2015.

[217] L. P. Kobbelt and S. Bischoff, "Parameterization-free active contour models with topology control," *The Visual Computer*, vol. 20, pp. 217–228, jun 2004.

[218] W. F. Noh and P. Woodward, "SLIC (Simple Line Interface Calculation)," in *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28 – July 2,*, (Twente University, Enschede), pp. 330–340, 1976.

[219] J. Li, "Calcul d'interface affine par morceaux," *Comptes rendus de l'Académie des sciences*, vol. 320, no. 8, pp. 391–396, 1995.

[220] J. López, J. Hernández, P. Gómez, and F. Faura, "A new volume conservation enforcement method for PLIC reconstruction in general convex grids," *Journal of Computational Physics*, vol. 316, pp. 338–359, jul 2016.

[221] J. López, J. Hernández, P. Gómez, and F. Faura, "Non-convex analytical and geometrical tools for volume truncation, initialization and conservation enforcement in VOF methods," *Journal of Computational Physics*, vol. 392, pp. 666–693, sep 2019.

[222] M.-H. Hsu and Y.-L. Hsu, "Interpreting three-dimensional structural topology optimization results," *Computers & Structures*, vol. 83, pp. 327–337, jan 2005.

[223] A. Nana, J.-C. Cuillière, and V. Francois, "Automatic reconstruction of beam structures from 3D topology optimization results," *Computers & Structures*, vol. 189, pp. 62–82, sep 2017.

[224] T. Zegard and G. H. Paulino, "Bridging topology optimization and additive manufacturing," *Structural and Multidisciplinary Optimization*, vol. 53, pp. 175–192, jan 2016.

[225] Y. Xian, D. W. Rosen, and G. W. Woodruff, "A post-processing procedure for level set based topology optimization," in *Solid Freeform Fabrication 2017: Proceedings of the 28th Annual International Solid Freeform Fabrication Symposium – An Additive Manufacturing Conference*, pp. 2201–2215, 2017.

[226] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, nov 1988.

[227] T. Chan and L. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[228] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillén-Aghion, J. C. Olivo-Marin, and C. Zimmer, "Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces," *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1396–1410, 2005.

[229] J. Sethian and A. Wiegmann, "Structural Boundary Design via Level Set and Immersed Interface Methods," *Journal of Computational Physics*, vol. 163, pp. 489–528, sep 2000.

[230] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, jan 1988.

[231] S. Bischoff and L. P. Kobbelt, "Snakes with topology control," *Image Rochester NY*, p. 10, 2003.

[232] A. Busemann, "Aerodynamic lift at supersonic speeds," *Luftfahrtforschung*, 1935.

[233] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming for large-scale nonlinear optimization," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 123–137, dec 2000.

[234] L. Piegl and W. Tiller, *The NURBS Book*. Springer Berlin Heidelberg, 1997.

[235] R. Grothmann and M. Sommer, "Composing circular arcs with minimal NURBS," *International mathematical forum*, vol. 4, no. 5-8, pp. 193–218, 2009.

[236] A. Jameson, S. Shankaran, and L. Martinelli, "Continuous Adjoint Method for Unstructured Grids," *AIAA Journal*, vol. 46, no. 5, pp. 1226–1240, 2008.

[237] I. Y. Kim and O. L. De Weck, "Variable chromosome length genetic algorithm for progressive refinement in topology optimization," *Structural and Multidisciplinary Optimization*, vol. 29, no. 6, pp. 445–456, 2005.

[238] K. Bandara, T. Ruberg, and F. Cirak, "Shape optimisation with multiresolution subdivision surfaces and immersed finite elements," *Computer Methods in Applied Mechanics and Engineering*, vol. 300, pp. 510–539, 2016.

[239] A. Jameson, W. Schmidt, and E. Turkel, "Numerical solution of the Euler equations by finite volume methods using Runge Kutta time stepping schemes," in *14th Fluid and Plasma Dynamics Conference*, (Reston, Virigina), American Institute of Aeronautics and Astronautics, jun 1981.

[240] P. Eliasson, *EDGE: A Navier-Stokes solver for unstructured grids*. 2001.

[241] J. Hall, T. Rendall, C. Allen, and H. Peel, "A multi-physics computational model of fuel sloshing effects on aeroelastic behaviour," *Journal of Fluids and Structures*, vol. 56, pp. 11–32, 2015.

[242] A. A. Demargne, R. Evans, P. Tiller, and W. N. Dawes, "Practical and Reliable Mesh Generation for Complex, Real-World Geometries," in *52nd Aerospace Sciences Meeting*, no. January, (Reston, Virginia), pp. 1–9, American Institute of Aeronautics and Astronautics, jan 2014.

[243] L. Kedward, C. Allen, and T. Rendall, "Efficient and exact mesh deformation using multiscale RBF interpolation," *Journal of Computational Physics*, vol. 345, pp. 732–751, sep 2017.

[244] F. Bisson and S. Nadarajah, "Adjoint-Based Aerodynamic Optimization of Benchmark Problems," in *53rd AIAA Aerospace Sciences Meeting*, no. January, (Reston, Virginia), pp. 1–20, American Institute of Aeronautics and Astronautics, jan 2015.

[245] W. R. Sears, "On projectiles of minimum wave drag," *Quart. Appl. Math*, vol. 4, no. 4, pp. 361–366, 1947.

[246] E. B. Klunker and K. Harder, "Comparison of Supersonic Minimum-Drag Airfoils Determined By Linear and Nonlinear Theory," Tech. Rep. February, National Aeronautics And Space Administration Hampton Va Langley Research Center, Washington, 1952.

[247] K. Palaniappan and A. Jameson, "Bodies Having Minimum Pressure Drag in Supersonic Flow: Investigating Nonlinear Effects," *Journal of Aircraft*, vol. 47, pp. 1451–1454, jul 2010.

[248] L. Kedward, A. D. Payot, T. Rendall, and C. B. Allen, "Efficient Multi-Resolution Approaches for Exploration of External Aerodynamic Shape and Topology," in *2018 Applied Aerodynamics Conference*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jun 2018.

[249] A. D. Payot, L. J. Kedward, T. Rendall, and C. B. Allen, "Optimisation of Multi-Modal Aerodynamic Shape and Topology Problems," in *AIAA Scitech 2019 Forum*, (Reston, Virginia), American Institute of Aeronautics and Astronautics, jan 2019.

[250] W. Haack, *Projectile Shapes for Smallest Wave Drag*. Graduate Division of Applied Mathematics, Brown University, 1948.

[251] A. Roshko and H. W. Liepmann, *Elements of gas dynamics*. Wiley, New York, 1957.

[252] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, vol. 47, no. 1, pp. 99–131, 2005.

[253] Z. Lyu, Z. Xu, and J. R. R. A. Martins, "Benchmarking Optimization Algorithms for Wing Aerodynamic Design Optimization," in *8th International Conference on Computational Fluid Dynamics (ICCFD8)*, (Chengdu, China), jul 2014.

[254] T. D. Economon, J. J. Alonso, T. A. Albring, and N. R. Gauger, "Adjoint Formulation Investigations of Benchmark Aerodynamic Design Cases in SU2," *35th AIAA Applied Aerodynamics Conference*, no. June, pp. 1–13, 2017.

[255] F. Palacios, M. R. Colonno, A. C. Aranake, A. Campos, S. R. Copeland, T. D. Economon, A. K. Lonkar, T. W. Lukaczyk, T. W. R. Taylor, and J. J. Alonso, "Stanford University Unstructured (SU2): An open-source integrated computational environment for multi-physics simulation and design," *51st AIAA Aerospace Sciences Meeting*, 2013.

[256] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation.," *Computational Geometry*, vol. 22, pp. 21–74, 2002.

[257] A. de Boer, M. S. van der Schoot, and H. Bijl, "Mesh deformation based on radial basis function interpolation," *Computers and Structures*, 2007.

[258] T. C. S. Rendall and C. B. Allen, "Reduced Surface Point Selection Options for Efficient Mesh Deformation using Radial Basis Functions," *Journal of Computational Physics*, 2010.

[259] J. K. Pugh, L. B. Soros, and K. O. Stanley, "Quality Diversity: A New Frontier for Evolutionary Computation," *Frontiers in Robotics and AI*, vol. 3, no. July, pp. 1–17, 2016.

[260] J. Pérez, "A New Golden Age of Minimal Surfaces," *Notices of the American Mathematical Society*, vol. 64, pp. 347–358, apr 2017.

[261] D. A. Burdette and J. R. Martins, "Design of a transonic wing with an adaptive morphing trailing edge via aerostructural optimization," *Aerospace Science and Technology*, vol. 81, pp. 192–203, oct 2018.

[262] S. Khosravi and D. W. Zingg, "Aerostructural Optimization of Drooped Wings," *Journal of Aircraft*, pp. 1–8, dec 2017.

[263] D. A. Masters, N. J. Taylor, T. Rendall, and C. B. Allen, "Three-Dimensional Subdivision Parameterisation for Aerodynamic Shape Optimisation," *55th AIAA Aerospace Sciences Meeting*, no. January, pp. 1–17, 2017.

[264] A. D. J. Payot, "RSVS3D." `https://github.com/payoto/rsvs3d`. [Online; accessed 08/06/2020].

[265] Eigen, "Dense Decomposition Benchmark." `https://eigen.tuxfamily.org/dox/group__DenseDecompositionBenchmark.html`. [Online; accessed 23/11/2019].

[266] Eigen, "Sparse Decomposition Tutorial." `https://eigen.tuxfamily.org/dox/group__TutorialSparse.html`. [Online; accessed 23/11/2019].

[267] G. Guennebaud, B. Jacob, *et al.*, "Eigen v3." http://eigen.tuxfamily.org, 2010.

[268] soapbubble.dk, "Catenoid Soap Bubble." `https://www.soapbubble.dk/en/articles/former`. [Online; accessed 28/11/2019].

[269] E. T. Institute, "Helicoidal Soap Bubble." `https://plus.maths.org/content/abel-prize-2019`. [Online; accessed 28/11/2019].

[270] H. C. Wente, "An existence theorem for surfaces of constant mean curvature," *Journal of Mathematical Analysis and Applications*, vol. 26, pp. 318–344, may 1969.

[271] E. L. Thomas, D. M. Anderson, C. S. Henkee, and D. Hoffman, "Periodic area-minimizing surfaces in block copolymers," *Nature*, vol. 334, pp. 598–601, aug 1988.

[272] H. Pan, Y.-K. Choi, Y. Liu, W. Hu, Q. Du, K. Polthier, C. Zhang, and W. Wang, "Robust modeling of constant mean curvature surfaces," *ACM Transactions on Graphics*, vol. 31, pp. 1–11, jul 2012.

[273] H. Wente, "Counterexample to a conjecture of H. Hopf," *Pacific Journal of Mathematics*, vol. 121, pp. 193–243, jan 1986.

[274] K. A. Brakke, "The Surface Evolver," *Experimental Mathematics*, vol. 1, pp. 141–165, jan 1992.

[275] K. A. Brakke, "The Surface Evolver and the stability of liquid surfaces," *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 354, pp. 2143–2157, sep 1996.

[276] D. Piker, "Kangaroo Physics." `https://www.food4rhino.com/app/kangaroo-physics`. [Online; accessed 28/11/2019].

[277] R. McNeel and Associates, "Rhinoceros 3D." `https://www.rhino3d.com/`. [Online; accessed 28/11/2019].

[278] H. Si, "TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator," *ACM Transactions on Mathematical Software*, vol. 41, pp. 1–36, feb 2015.

[279] N. Ames, "CART3D." `https://www.nasa.gov/ames-partnerships/technology/technology-opportunity-cart3d`. [Online; accessed 01/12/2019].

[280] A. Technologies, "Viscous CART3D." `https://www.aerion-tech.com/cart3dv-docs/`. [Online; accessed 01/12/2019].

[281] NASA, "Common Research Model." `https://commonresearchmodel.larc.nasa.gov/geometry/stp-files/`. [Online; accessed 03/12/2019].

[282] Eigen, "Multi Threading." `https://eigen.tuxfamily.org/dox/TopicMultiThreading.html`. [Online; accessed 23/11/2019].

[283] R. A. Canfield and M. S. Eldred, "Quadratic multipoint exponential approximation for optimization and uncertainty quantification," *AIAA Scitech 2019 Forum*, no. January, pp. 1–8, 2019.

[284] M. Sabin, "Eigenanalysis and Artifacts of Subdivision Curves and Surfaces," in *Tutorials on multiresolution in geometric modelling : summer school lectures notes*, pp. 69–92, 2002.

[285] P. Shepherd and P. Richens, "The case for Subdivision Surfaces in building design," *Journal of the International Association for Shell and Spatial Structures*, vol. 53, no. 174, pp. 237–245, 2012.

[286] P. B. Bornemann and F. Cirak, "A subdivision-based implementation of the hierarchical b-spline finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 253, pp. 584–598, 2013.